DOI: 10.35191/medsoft_2026_1_38_kofranek_ll

SVELTE BODYLIGHT: NOVÁ TECHNOLOGIE TVORBY WEBOVÝCH SIMULÁTORŮ

Jiří Kofránek, Eduard Strauss, Michail Pugachov

Abstrakt

Navazujeme na technologii Bodylight.js, se kterou jsme od roku 2019 vytvářeli webové simulátory pro výuku medicíny. Nová generace, Svelte Bodylight, zachovává její principy: model se počítá v internetovém prohlížeči, bez serveru a bez instalace, a aplikace se skládá ze standardizovaných webových komponent. Mění se způsob, jakým aplikace vzniká. Modely z jazyka Modelica se přes standard FMI překládají do WebAssembly z Dymoly i z OpenModeliky a každý převod se automaticky porovnává s referenčním výpočtem. Uživatelské rozhraní vzniká v open-source překladači Svelte jako sada webových komponent: posuvníky, grafy průběhů i tlakově-objemových smyček, kontrolky stavů a řízené scénáře. Pedagog sestaví aplikaci v konfigurátoru bez programátora a hotový simulátor se doručí jako jediný soubor HTML, který běží i bez připojení k internetu.

Klíčová slova

Modelica, FMI, WebAssembly, Svelte, webové komponenty, webové simulátory, lékařská výuka

1 Úvod: co z Bodylight.js zůstává a co se mění

Lékařský simulátor spojuje výklad s matematickým modelem. Student mění parametry, sleduje, jak na to reaguje model fyziologického systému, a z chování modelu pochopí, proč se nemoc projevuje tak, jak se projevuje. Tvorba takového simulátoru je ale týmová práce. V článku z konference MEDSOFT 2024 [1.] jsme popsali čtyři role, které se na ní podílejí: tvůrce modelů, výtvarník, pedagog a programátor. Tvůrce modelů napíše model v jazyce Modelica, výtvarník nakreslí animace, pedagog navrhne scénář výuky a programátor z toho všeho složí aplikaci (obr. 1 vlevo).

Pro poslední krok jsme od roku 2019 používali technologii Bodylight.js [2.–6.]. Model se v ní překládá do WebAssembly a počítá přímo v prohlížeči, takže simulátor nepotřebuje server ani instalaci, a aplikace se skládá ze standardizovaných webových komponent. To všechno platí i nadále. Úzkým hrdlem ale zůstával programátor: každou aplikaci skládal ručně, a každá změna scénáře tak čekala na něj.

Svelte Bodylight tuto roli přesouvá. Aplikaci skládá pedagog v konfigurátoru, a programátor vytváří jen nové druhy komponent – jednou, pro všechny aplikace (obr. 1 vpravo). Tento článek popisuje, jak to funguje: cestu modelu z Modeliky do prohlížeče, ověření, že simulátor počítá totéž co původní program, komponenty uživatelského rozhraní, experiment jako popis místo programu, konfigurátor a doručení celé sady simulátorů jako jediného souboru. Simulátory samotné v provozu ukazuje druhý příspěvek tohoto sborníku [7.].

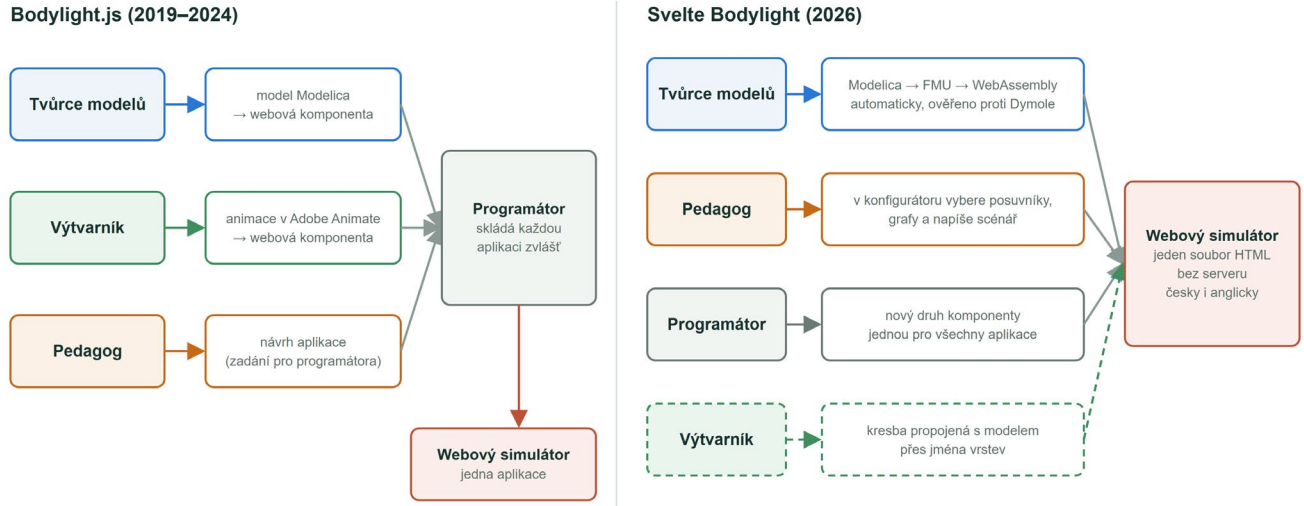
2 Od Modeliky do prohlížeče

Modely fyziologických systémů píšeme v jazyce Modelica [8.]. Je to jazyk akauzální: modelář zapisuje rovnice a vztahy mezi veličinami tak, jak je zná z fyziologie, a o tom, v jakém pořadí a kterou proměnnou z které rovnice počítat, rozhoduje překladač. Z modelu se pak vyrobí FMU (Functional Mock-up Unit) podle standardu FMI 2.0 [9.] – samostatná výpočetní jednotka, kterou může spouštět jakýkoli program, jenž standard podporuje. Používáme variantu Co-Simulation, ve které FMU nese vlastní řešič diferenciálních rovnic (CVODE z knihovny SUNDIALS [10.]) a navenek se jen posouvá v čase po krocích. FMU musí obsahovat zdrojový kód v jazyce C, protože ten pak přeložíme pro prohlížeč.

Překlad do prohlížeče obstarává Emscripten [11.], překladač z jazyka C do WebAssembly [12.] – binárního formátu, který dnešní prohlížeče spouštějí rychlostí blízkou nativnímu kódu. Výsledkem jsou dva soubory: modul WebAssembly s modelem a malý soubor JavaScriptu, který ho zavede. Dvacet sekund simulace krevního oběhu, tedy 5 000 kroků, se spočítá zhruba za desetinu sekundy; zbytek času patří kreslení grafů. Celý řetěz shrnuje obr. 2.

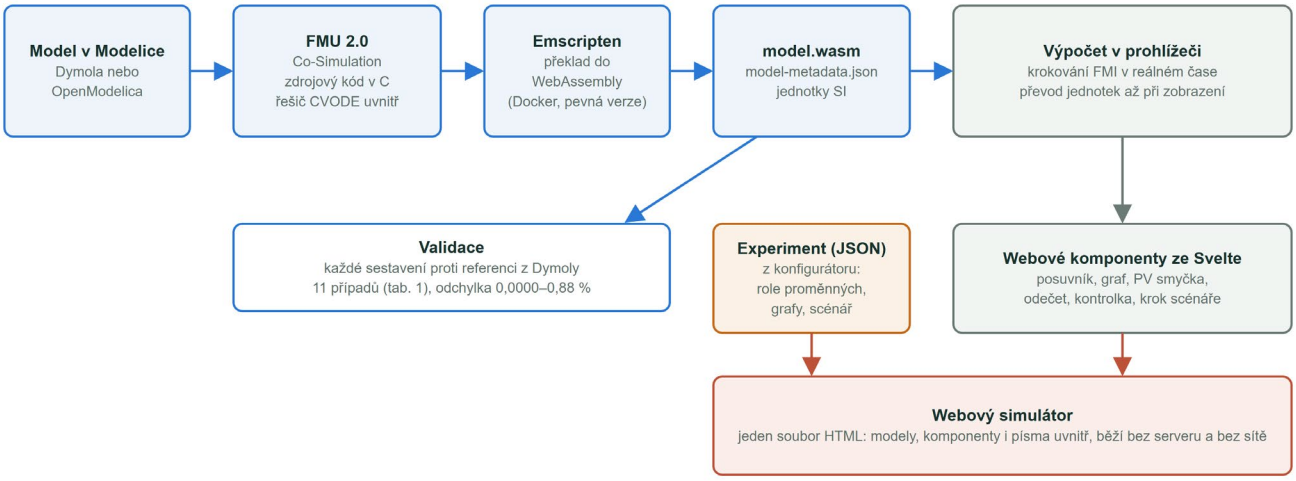
FMU přijímáme ze dvou zdrojů. Prvním je Dymola, ve které vznikla většina našich modelů. Druhým je OpenModelica [13.], volně dostupné prostředí; cílem je, aby celý katalog šel postavit i bez licence Dymoly. Oba zdroje se chovají trochu jinak – OpenModelica například řeší události (otevření chlopně) přesně v okamžicích, kdy ji hostitel posune v čase, takže pulzující model potřebuje jemnější krok. Jemnost krokování je proto vlastností modelu, kterou nese popis experimentu, a ne ovládacím prvkem pro studenta.

Uvnitř modelu je všechno v jednotkách SI: tlak v pascalech, objem v krychlových metrech, průtok v krychlových metrech za sekundu.



Obrázek 1 – role při tvorbě webového simulátoru. Vlevo technologie Bodylight.js (2019–2024): tvůrce modelů, výtvarník i pedagog předávají svou práci programátorovi, který z ní skládá každou aplikaci zvlášť, a pedagogův záměr se do aplikace dostane jen přes něj. Vpravo Svelte Bodylight (2026): aplikaci skládá pedagog v konfigurátoru a programátor rozšiřuje sadu komponent, jednou pro všechny aplikace. Čárkovaně je vyznačen další krok, který zatím není implementován – kresba výtvarníka propojená s modelem.

DOI: 10.35191/medsoft_2026_1_38_kofranek_ll



Obrázek 2 – od modelu v Modelice ke stránce v prohlížeči. Modře cesta modelu: z Dymoly nebo OpenModeliky jako FMU se zdrojovým kódem v C, překlad Emscriptenem do WebAssembly a automatické ověření proti referenčnímu výpočtu v Dymole. Šedě výpočet v prohlížeči a webové komponenty, společné pro všechny modely. Oranžově experiment, o kterém rozhoduje pedagog. Červeně výsledek, který dostane student: jeden soubor HTML, který běží bez serveru a bez sítě.

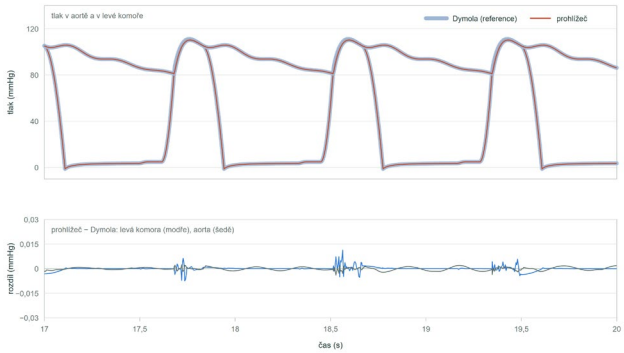
Převod na jednotky, ve kterých čte lékař – mmHg, ml, l/min –, se dělá až při zobrazení. Model tak zůstává fyzikálně konzistentní a jednotky jsou věcí rozhraní, ne rovnic.

Překladový řetěz běží v kontejneru Docker s pevně zvolenou verzí překladače Emscripten. Stejný model tak dá stejný výsledek na kterémkoli počítači a za rok stejný jako dnes – což je předpoklad toho, aby mělo smysl ho ověřovat.

3 Ověření: simulátor počítá totéž co Dymola

Převod modelu má mnoho kroků a v každém se může něco ztratit. Proto se každý převedený model automaticky porovnává s referenčním výpočtem v Dymole: tentýž scénář se spočítá v prohlížečovém modelu a výsledné průběhy se porovnají s průběhy z Dymoly bod po bodu. Porovnání je součástí každého sestavení; změna, která by výpočet pokazila, neprojde.

U většiny veličin se odchylka vztahuje k okamžité hodnotě reference. U pulzujících signálů to nejde: tlak v komoře prochází během každého stahu téměř nulou, a malá absolutní chyba by tam dala obrovskou relativní. Odchylku pulzujících signálů proto vztahujeme k rozsahu signálu (v tabulce „k rozsahu“).



Obrázek 3 – ověření simulátoru proti Dymole na modelu krevního oběhu, poslední tři sekundy dvacetisekundového běhu. Nahoře tlak v aortě a v levé komoře: široká světlá čára je referenční výpočet v Dymole, tenká tmavá čára simulátor v prohlížeči; obě se v celém průběhu překrývají. Dole jejich rozdíl: v zobrazených třech sekundách nejvýš 0,011 mmHg v levé komoře a 0,004 mmHg v aortě (za celý běh 0,024 a 0,007 mmHg); soustředí se do okamžiků otevírání a zavírání chlopní.

model a scénář	největší odchylka	tolerance
jednoduchá respirace, konstantní vstupy, 60 s	0,0028 %	0,05 %
jednoduchá respirace, skoková změna vstupu	0,27 %	0,5 %
normální krevní oběh, Dymola	0,0216 % k rozsahu	0,025 %
normální krevní oběh, OpenModelica (původní model)	0,0086 % k rozsahu	0,02 %
normální krevní oběh, OpenModelica (balík v jednotkách SI)	0,0117 % k rozsahu	0,02 %
respirace z knihovny Bodylight, OpenModelica	0,0025 %	0,05 %
jednoduchá respirace z diagramu OPM	0,0026 %	0,05 %
normální krevní oběh z diagramu OPM	0,0157 % k rozsahu	0,02 %
Hodgkinův–Huxleyho model na prahu podráždění	0,88 % k rozsahu	1 %
metabolismus železa, ustálený stav	0,0000 %	0,05 %
metabolismus železa, vyřazený gen pro hepcidin	0,12 % k rozsahu	0,5 %
metabolismus železa, zánět po podání lipopolysacharidu	0,013 % k rozsahu	0,1 %

Tabulka 1 – porovnání simulátorů v prohlížeči s referenčním výpočtem v Dymole. Jak malé ty odchylky jsou, ukazuje obr. 3 na modelu krevního oběhu: tlak v levé komoře a v aortě z Dymoly a ze simulátoru v prohlížeči se v grafu úplně překrývají a jejich rozdíl nepřesáhne 0,025 mmHg – zlomek toho, co by šlo vyčíst z jakéhokoli monitoru.

Za komentář stojí jedna položka. Model nervové membrány je porovnáván schválně na samém prahu podráždění: podnět 18 mV membránu jen prohne, 19 mV vyvolá akční potenciál, a případ tak dokazuje, že práh leží přesně tam, kde v Dymole. Je to nejcitlivější možné místo a odchylka 0,88 % se týká okamžiku, kdy se potenciál mění nejrychleji.

DOI: 10.35191/medsoft_2026_1_38_kofranek_ll

Porovnání odhalilo i chyby, které by jinak zůstaly skryté. Jedna z nich způsobila, že se model po určité době tiše přestal posouvat v čase, jiná, že aplikace zapsala nastavené hodnoty do modelu jen před jeho inicializací a model pak běžel s výchozími hodnotami. Obě pohledem na grafy vidět nebyly; obě zachytilo číselné porovnání. U modelu metabolismu železa porovnání ukázalo, že Dymola i prohlížeč počítaly ubývající lipopolysacharid nepřesně, každý jinak: model ho počítal jako rozdíl dvou téměř stejných čísel. Po přeformulování modelu se shodují na 0,013 %.

4 Webové komponenty ze Svelte

Uživatelské rozhraní stavíme pomocí Svelte [14.], open-source překladače komponent. Svelte na rozdíl od běžných knihoven nepřidává do stránky rozsáhlý běhový systém; komponenty přeloží do úsporného kódu, který přímo upravuje stránku. Z komponent pak vznikají standardní webové komponenty (custom elements) [15.] – vlastní značky HTML, které lze vložit do libovolné webové stránky, bez ohledu na to, v čem je napsán zbytek stránky.

Sada má dnes třináct komponent. Nejdůležitější jsou:

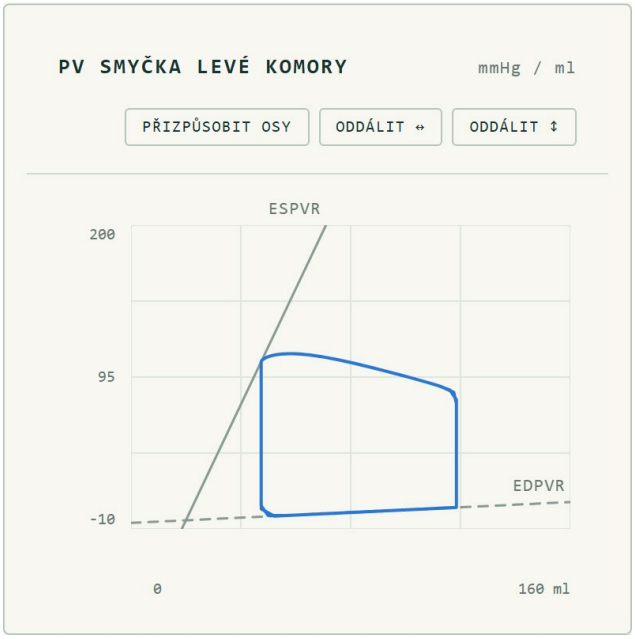
- **posuvník s číselným polem** (<fmu-slider>): mění vstup modelu; hodnotu lze i napsat, a pokud se posuvník pohnul, nabídne cestu zpět k výchozí hodnotě;
- **graf** (<fmu-chart>): průběh v čase s oknem historie jako na monitoru u lůžka, nebo graf hodnoty proti hodnotě – například tlakově-objemová smyčka levé komory s přímkami ESPVR a EDPVR (obr. 4);
- **dlaždice s odečtem** (<fmu-readout>): číslo s jednotkou, jak ho ukazuje monitor;
- **kontrolka stavu a tlačítko** (<fmu-state>, <fmu-trigger>): pro kvalitativní modely, ve kterých se nemění čísla, ale stavy – kontrolka svítí, když systém je v daném stavu, tlačítko spustí událost;
- **krok scénáře** (<fmu-step>): text pro studenta, tlačítka Zpět a Dále a tlačítko, které nastaví hodnoty kroku.

Zbylé komponenty ovládají běh (spustit, pozastavit, nový běh), ukazují stav a čas simulace a spojují všechno s jedním modelem (<fmu-experiment>). Podstatné je, že každá komponenta existuje jen jednou. Táž sada slouží aplikaci, konfigurátoru i samostatné stránce, takže se nemohou rozejít – oprava v jednom místě platí všude.

Simulace samotná do komponent nepatří. Pořadí volání standardu FMI, krokování v čase, udržování tempa a převod jednotek má na starosti jediná vrstva bez vazby na rozhraní, kterou používají všechny stránky. V aplikaci se model počítá ve vlákně Web Worker, aby nebrzdil rozhraní; v samostatné stránce na hlavním vlákně, protože prostředí, ve kterých se stránka sdílí, mohou samostatné vlákno zakázat. Vrstva proto hlídá, aby výpočet i kreslení držely slíbené tempo i na pomalejším počítači: když kreslení grafů zdražuje, kreslí se řidčeji, ale simulace běží dál v reálném čase.

5 Experiment jako popis, ne jako program

Aplikaci nad modelem neprogramujeme, ale popisujeme. Popis se jmenuje experiment a je to soubor ve formátu JSON. Říká, které proměnné modelu jsou posuvníky a v jakém rozsahu, které jsou pevně nastavené parametry, které se jen zobrazují a v jakých jednotkách, jaké grafy se kreslí a z jakých křivek, jak dlouho



Obrázek 4 – tlakově-objemová smyčka levé komory v simulátoru normálního krevního oběhu. Šikmá plná přímka je ESPVR (vztah tlaku a objemu na konci systoly), čárkovaná křivka EDPVR (vztah tlaku a objemu na konci diastoly). Tlačítky nad grafem se osy přizpůsobí právě zobrazené smyčce nebo se oddálí, aby se při změně parametrů dalo sledovat, kam se smyčka posouvá.

a jak rychle simulace běží a jaký je scénář. Model sám se kvůli experimentu nemění; nad týmž modelem jich může být libovolně mnoho. Nad modelem krevního oběhu jich dnes máme osm – od normálního oběhu přes selhání komor, chlopenní vady a krvácení až po nitrobřišní tlak a protišokové kalhoty.

Scénář je řada kroků. Každý krok má text pro studenta, hodnoty, které nastaví, a údaj, zda začíná znovu – jako nový pacient –, nebo pokračuje tam, kde předchozí skončil, jako u krvácení, které postupuje. Čísla v textech scénářů nejsou odhadnutá: jsou změřená na běžícím modelu a zaokrouhlená tak, jak je ukáže stránka, a nástroj je porovnává s tím, co student na stránce skutečně uvidí. Tak se mimo jiné našly kroky, které měly začínat znovu, ale pokračovaly z konce předchozího kroku, a ukazovaly proto jiná čísla, než sliboval text.

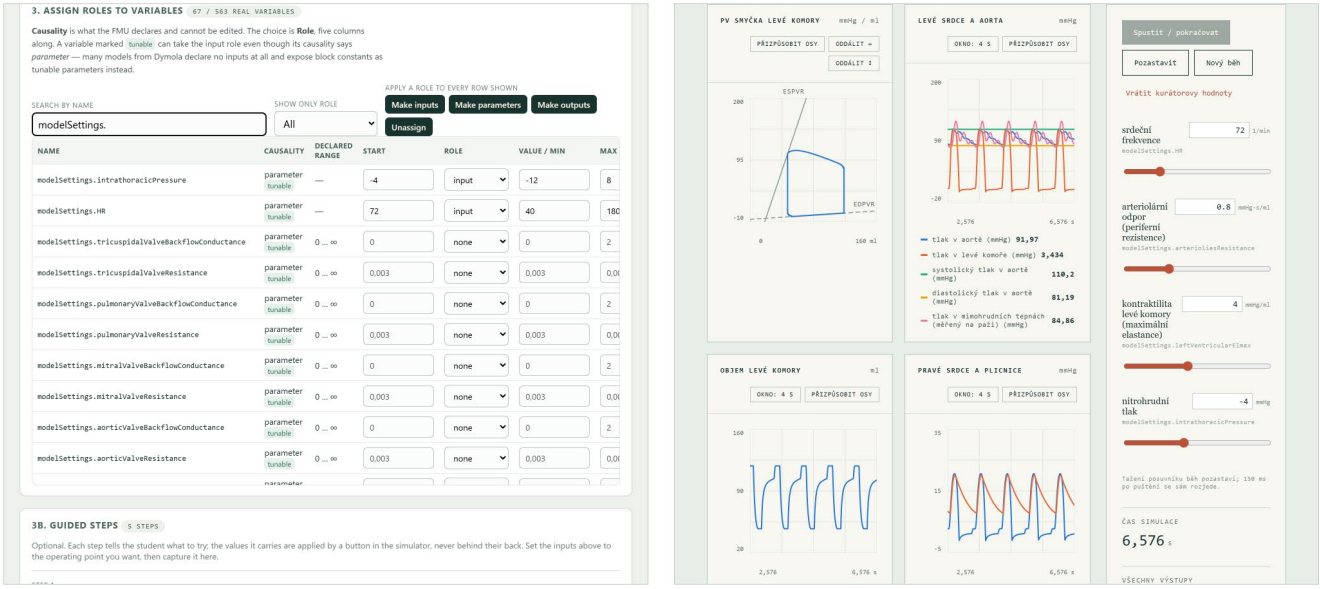
Jeden experiment nese obě jazykové verze. Text je uložen česky i anglicky, ale čísla, rozsahy a nastavení jsou společné, takže se obě verze nemohou rozejít ve věcném obsahu. Česká stránka píše desetinnou čárku, anglická tečku; automatická kontrola ověřuje, že se obě verze téže stránky liší jen slovy.

Vedle obecné stránky, kterou aplikace z experimentu sestaví sama, může mít experiment i navrženou stránku – soubor HTML, ve kterém výtvarník nebo pedagog rozmístí prvky podle svého a jen je označí jmény proměnných. Tak vznikl například „kokpit“ respirace nebo živý diagram, ve kterém se hodnoty modelu zobrazují přímo v uzlech schématu, nebo schéma metabolismu železa nakreslené podle náčrtu autora modelu, s živými hodnotami u orgánů i u míst, kde hepcidin a interleukin 6 blokují feroportin.

6 Konfigurátor: aplikaci skládá pedagog

Experiment se nepíše ručně. Pedagog ho sestaví v konfigurátoru: načte popis modelu, u každé proměnné zvolí, zda bude posuvníkem, pevným parametrem, nebo zobrazeným výstupem, vybere jednotky, rozdělí výstupy do grafů, nastaví délku a tempo simulace a napíše scénář. Konfigurátor přitom nemá vlastní simulaci – řídí skutečnou aplikaci v sousedním okně, tak-

DOI: 10.35191/medsoft_2026_1_38_kofranek_ll



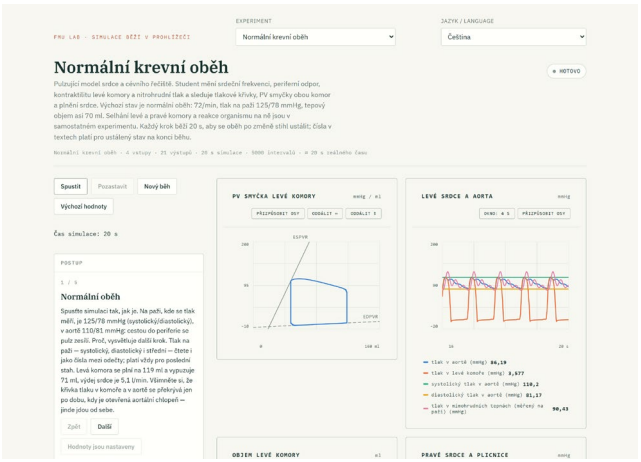
Obrázek 5 – konfigurátor (vlevo) a aplikace, kterou řídí (vpravo). V konfigurátoru pedagog přiřazuje proměnným modelu krevního oběhu role – vstup ovládaný posuvníkem, pevný parametr, zobrazený výstup – a nastavuje jejich rozsahy; vpravo běží skutečná aplikace s grafy a posuvníky, které z těchto voleb vznikly. Rozhraní konfigurátoru je zatím anglické.

že pedagog vidí přesně to, co pak uvidí student, a zkušebním během ověří, že zvolené tempo počítač stíhá (obr. 5).

Konfigurátor také hlídá chyby, které by se jinak projevily až ve výuce: proměnnou, kterou model nemá, vstup, který nejde měnit, nebo výchozí hodnotu posuvníku, která neleží na jeho stupnici.

7 Doručení: jeden soubor bez serveru

Celá sada simulátorů se sestaví do jediného souboru HTML. Uvnitř jsou modely ve WebAssembly, komponenty, popisy experimentů i písma; soubor má kolem 6,5 MB a funguje bez připojení k internetu – stačí ho otevřít v prohlížeči, například v posluchárně bez sítě, nebo uložit na libovolný webový server. Dnes obsahuje 16 modelů a 23 experimentů (obr. 6).



Obrázek 6 – sdílená stránka se simulátorem normálního krevního oběhu po dokončeném běhu: výběr experimentu a jazyka, popis, ovládání běhu, první krok scénáře a grafy – tlakově-objemová smyčka levé komory a tlaky v levém srdci a v aortě s odečty. Celá sada simulátorů je jediný soubor HTML, který funguje i bez připojení k internetu.

Stránku hlídá řada automatických kontrol. Ověřují, že se nikam nepřipojuje a že písma bere ze sebe, že každý experiment něco ukáže a reaguje na ovládání, že se obě jazykové verze liší jen slovy, že stránka funguje v prohlížečích Chrome a Firefox i v jádře WebKit, na kterém stojí Safari, že se vejde na displej mobilního telefonu a že experimenty drží slíbené tempo i na počítači dvakrát pomalejším, než na jakém vznikly. Poslední dvě kontroly přibyly, když se ukázalo, že stránka se na telefonu vykreslovala zmenšená na třetinu a že na pomalejším počítači mohl model krevního oběhu běžet několikanásobně pomaleji, než sliboval – obojí na vývojovém počítači vidět nebylo.

8 Odkazy na běžící simulátory

Stejně jako v článku z roku 2024 si čtenář může simulátory vyzkoušet sám: odkaz otevře simulátor v jeho prohlížeči. Odkaz umí otevřít přímo jeden experiment a jazyk – například *https://bodylight.physiome.cz/SvelteBodylight.js/medsoft2026/fmu-lab.html?experiment=normal-circulation/krvaceni&lang=cs* otevře scénář krvácení česky; s &lang=en anglicky a s &view=opd u modelu z diagramu OPM rovnou diagram. Stránka sama udržuje v adresním řádku, co je právě zobrazeno, takže odkaz na kterýkoli pohled lze jednoduše zkopírovat. Odkazy na všechny experimenty jsou v příspěvku [7.]. Přehled všech simulátorů s odkazy je na *https://bodylight.physiome.cz/SvelteBodylight.js/*.

9 Další krok: výtvarník a modely z diagramů

Ve Svelte Bodylight zatím chybí role, která byla v Bodylight.js nejviditelnější: výtvarník. Navrhujeme, aby kresba vytvořená v nástroji, který výtvarník zná – Adobe Animate, Illustrator nebo After Effects –, nesla jména vrstev a instancí, a ta se v konfigurátoru propojila s proměnnými modelu (obr. 7). Objem komory by pak vybíral snímek animace stahu srdce, chlopeň by se otevírala s průtokem a barva krve by sledovala saturaci. A naopak: klepnutí na chlopeň by mohlo nastavit její stenózu. Výtvarník by nepsal žádný kód; smlouvou mezi kresbou a modelem by byla jména vrstev. Tato část zatím není implementována.

Tentýž řetěz už dnes simuluje i modely, které nevznikly psaním v Modelice, ale z diagramů objektově procesní metodiky (OPM, norma ISO 19450 [16.]). Z diagramu se vygeneruje model v Modelice a ten projde stejnou cestou do prohlížeče; v tabulce 1 jsou dva takové modely a jejich odchylky od referencí jsou stejně malé jako u modelů psaných ručně. Podrobnosti budou předmětem samostatné práce.



Obrázek 7 – navržený další krok: kresba výtvarníka řízená modelem. Kresba z nástroje Adobe Animate, Illustrator nebo After Effects nese pojmenované vrstvy a instance, které se v konfigurátoru propojí s proměnnými modelu. Model pak řídí kresbu (objem komory vybírá snímek animace, chlopeň se otevírá s průtokem) a kresba ovládá model (klepnutí na chlopeň nastaví stenózu). Výtvarník nic neprogramuje; smlouvou mezi kresbou a modelem jsou jména vrstev. Zatím neimplementováno.

10 Závěr

Svelte Bodylight zachovává to, co se na Bodylight.js osvědčilo – standardy Modelica, FMI a webové komponenty, výpočet v prohlížeči bez serveru a týmovou tvorbu –, a mění to, co bylo úzkým hrdlem. Aplikaci nad modelem dnes neskládá programátor, ale pedagog, a to v konfigurátoru, který mu ukazuje skutečnou běžící aplikaci. Každý převedený model se automaticky ověřuje proti původnímu výpočtu a hotová sada simulátorů je jediný soubor, který funguje i bez sítě. Cesta od modelu k výukové aplikaci se tím zkracuje z týdnů na hodiny – simulátor metabolismu železa měl první experiment s ovládáním hepcidinu hotový týž den, kdy byl model dodán – a přesouvá se k tomu, kdo ví, co chce studenty naučit. Jak takové simulátory vypadají ve výuce, ukazuje druhý příspěvek tohoto sborníku [7.].

Poděkování

Projekt byl financován společností Creative Connections s. r. o. a Grantovou agenturou UK.

SVELTE BODYLIGHT: A NEW TECHNOLOGY FOR BUILDING WEB SIMULATORS

Abstract

We build on Bodylight.js, the technology with which we have been creating web simulators for medical education since 2019. The new generation, Svelte Bodylight, keeps its principles: the model is computed in the web browser, with no server and no installation, and the application is assembled from standardised web components. What changes is how the application comes into being. Models written in Modelica are translated through the FMI standard into WebAssembly from Dymola as well as from OpenModelica, and every translation is compared automatically with a reference computation. The user interface is built with the open-source compiler Svelte as a set of web components: sliders, charts of time courses and of pressure–volume loops, state lamps and guided scenarios. A teacher assembles the application in a configurator without a programmer, and the finished simulator is delivered as a single HTML file that runs even without an internet connection.

Keywords

Modelica, FMI, WebAssembly, Svelte, web components, web simulators, medical education

Literatura

[1.] Kofránek J, Kulhánek T. Vysvětlující počítačové modely (explanatory models) v lékařské výuce. Medsoft. 2024;36(1): 1–14. doi:10.35191/medsoft_2024_1_36_kofranek

[2.] Šilar J, Polák D, Mládek A, Ježek F, Kurtz TW, DiCarlo SE, et al. Development of in-browser simulators for medical education: introduction of a novel software toolchain. J Med Internet Res. 2019;21(7): e14160. doi:10.2196/14160

[3.] Polák D, Ježek F, Šilar J, Kofránek J. Technologie tvorby webových simulátorů. Medsoft. 2019;31: 122–139. Získáno: https://www.medsoft.website/sbornik/2019/Medsoft_2019_Polak_Print.pdf

[4.] Kulhánek T, Mládek A, Brož M, Kofránek J. Bodylight.js web components – webové komponenty pro webové simulátory. Medsoft. 2021;33(1): 48–52. doi:10.35191/medsoft_2021_1_33_48_52

[5.] Kofránek J, Kulhánek T. Standardizace – cesta k open source technologiím pro webové simulátory. Medsoft. 2021;33(1): 25–34. doi:10.35191/medsoft_2021_1_33_25_34

[6.] Kulhánek T, Mládek A, Kofránek J, Ježek F. Bodylight.js 2.0 – Web components for FMU simulation, visualisation and animation in standard web browser. In: Proceedings of the 15th International Modelica Conference 2023, Aachen, October 9–11. Linköping Electronic Conference Proceedings 204; 2023. s. 443–452. doi:10.3384/ecp204443

[7.] Strauss E, Pugachov M, Kulhánek T, Kofránek J. Webové simulátory při výuce medicíny. Medsoft. 2026. (tento sborník)

[8.] Modelica Association. Modelica – A Unified Object-Oriented Language for Systems Modeling. Language Specification, version 3. 6. 2023. Získáno: https://specification.modelica.org/maint/3.6/MLS.html

[9.] Modelica Association. Functional Mock-up Interface for Model Exchange and Co-Simulation, version 2. 0. 2014. Získáno: https://fmi-standard.org

[10.] Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, et al. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Softw. 2005;31(3): 363–396. doi:10.1145/1089014.1089020

[11.] Zakai A. Emscripten: an LLVM-to-JavaScript compiler. In: Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications. 2011. s. 301–312. doi:10.1145/2048147.2048224

[12.] Haas A, Rossberg A, Schuff DL, Titzer BL, Holman M, Gohman D, et al. Bringing the web up to speed with WebAssembly. In: Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation. 2017. s. 185–200. doi:10.1145/3062341.3062363

[13.] Fritzson P, Pop A, Abdelhak K, Ashgar A, Bachmann B, Braun W, et al. The OpenModelica integrated environment for modeling, simulation, and model-based development. Model Identif Control. 2020;41(4): 241–295. doi:10.4173/mic.2020.4.1

[14.] Svelte. Získáno: https://svelte.dev

[15.] WHATWG. HTML Living Standard – Custom elements. Získáno: https://html.spec.whatwg.org/multipage/custom-elements.html

[16.] ISO 19450:2024. Automation systems and integration – Object-Process Methodology. Geneva: ISO; 2024.

Kontakt

doc. MUDr. Jiří Kofránek, CSc.
Oddělení biokybernetiky
Ústav patologické fyziologie 1. LF UK
U nemocnice 5, 128 53 Praha 2
+420 777 68 68 68
kofranek@gmail.com

MUDr. Eduard Strauss
Oddělení biokybernetiky
Ústav patologické fyziologie 1. LF UK
U nemocnice 5, 128 53 Praha 2
strausseda@gmail.com

Mgr. Michail Pugachov
Oddělení biokybernetiky
Ústav patologické fyziologie 1. LF UK
U nemocnice 5, 128 53 Praha 2
mpugachov@gmail.com