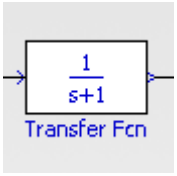
Portál pre odborné publikovanie ISSN 1338-0087

Internetom podporovaný návrh blokových schém v Matlab/Simulinku

Janík Zoltán · Elektrotechnika, Študentské práce

07.06.2010



V tejto práci je predstavený nový on-line nástroj umožňujúci vytváranie blokových schém Matlab/Simulink v internetovom prostredí. Návrh blokových schém je riešený podobným spôsobom ako pri lokálnej inštalácii Matlabu. Celá aplikácia je vytvorená v rozšírených technológiách ako je XHTML, CSS, JavaScript, PHP a AJAX. Vhodné využitie aplikácie je napríklad ako podporný nástroj vo virtuálnych a vzdialených (remote) laboratóriách.

1. Úvod

Rozšírenie internetu prináša nové možnosti do takmer každej oblasti každodenného života. Oblasť vzdelávania nie je výnimkou. Dôležitosť virtuálnych a vzdialených laboratórií sa stáva významným hlavne v oblasti vzdelania technického smeru. Virtuálne laboratória pomáhajú študentom pochopiť základné úlohy, pretože popisujú a zobrazujú riadený systém, umožňujú precvičiť si každý krok návrhu počnúc identifikáciou sústavy a končiac vyhodnotením výsledkov riadenia dosiahnutých konkrétnym modelom.

Výhodou takýchto laboratórií je ich prístupnosť 24 hodín denne z akéhokoľvek miesta. Študenti a zainteresovaní používatelia k nim môžu pristupovať cez bežne používané internetové prehliadače z akéhokoľvek počítača pripojeného na internet. V tejto práci je opísané nové rozšírenie funkčnosti virtuálnych laboratórií z oblasti automatizácie a riadenia – internetová aplikácia schopná editovať blokové schémy, ktoré sú použité v on-line experimentoch.

Vytvorený nástroj podporuje návrh simulácií v on-line laboratóriách vytvorených za podpory prostredia Matlab/Simulink. Samotná simulácia sa vykonáva na základe blokovej schémy, ktorá korešponduje s riadením virtuálneho alebo vzdialeného zariadenia. Je potrebné, aby mohli študenti nielen spustiť simuláciu s prednastavenými parametrami, ale aj zmeniť parametre riadenia alebo simulácie alebo dokonca zmeniť celý riadiaci algoritmus. Takýto prístup im umožní lepšie pochopenie všetkých aspektov úlohy riadenia. Väčšina internetom podporovaných riešení umožňuje modifikácie jedným z nasledujúcich spôsobov:

- použitím štandardného webového formulára, ktorý môže zahŕňať parametre riadenia/simulácie alebo kód s riadiacim algoritmom

- uploadom celého súboru, ktorý obsahuje parametre riadenia alebo celý riadiaci algoritmus
- parsovaním textového súboru, ktorý obsahuje riadiaci algoritmus. V tomto prípade sú zmeny umožnené len určitých riadkov súboru

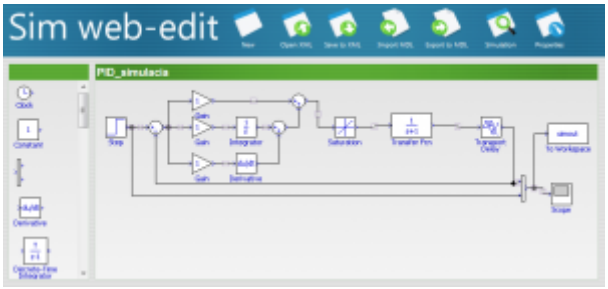
Vyvinutá aplikácia ponúka alternatívny spôsob oproti týmto riešeniam. Umožňuje vytvoriť schému riadenia pohodlným spôsobom cez webové rozhranie jednoduchým štýlom „drag & drop“ a prepájaním grafických blokov. Takýto prístup je typický pre desktopové aplikácie ako je napríklad Matlab/Simulink alebo Scilab/Scicos. Výhoda riešenia spočíva v tom, že zmeny algoritmu riadenia sú prostredníctvom tejto aplikácie realizované jednoduchými úpravami blokovej schémy, t.j. zmenou, vymazaním a pridávaním blokov. Aplikácia je bližšie popísaná v nasledujúcich kapitolách.

2. Popis aplikácie

2.1 Vzhľad

Front-end aplikácie pozostáva z troch častí (obr.1). Panel nástrojov (toolbar) obsahuje tlačidlá na vytvorenie, načítanie a uloženie schémy a na nastavenie parametrov blokovej schémy. V ľavej časti sa nachádza panel so zoznamom všetkých dostupných blokov. Hlavnou časťou je kresliace plátno s aktuálnou blokovou schémou. Pri vytváraní blokovej schémy je najskôr potrebné zobraziť blok alebo spojenie na plátne.

Zobrazenie je realizované prostredníctvom XHTML elementov (, <img>,...) s konkrétnym obrázkom bloku alebo prepojovacej úsečky. Okrem vizualizácie sú pre každý blok vytvárané aj inštancie JavaScriptových objektov. Jednotlivé inštancie nesú informácie o špecifických vlastnostiach (napríklad pozícia, rotácia, atribúty bloku, parametre prepojenia, ...).



Obr. 1. Vytvorenie blokovej schémy

2.2 Vkladanie blokov do schémy

Panel s blokmi obsahuje všetky dostupné bloky vo forme, v akej sa zobrazujú na plátne. Každý jeden blok má nastavenú možnosť presúvania myšou (draggable). Po tom, ako začne používateľ presúvať blok z panela, je vytvorený klon zvoleného bloku, ktorý môže byť následne vložený do plátna so schémou. Nový blok je vložený presne na to miesto, nad ktorým používateľ uvoľnil tlačidlo myši. V prípade, že sa používateľ pokúsi vložiť blok na iné miesto ako do kresliaceho plátna, nie je to považované za platnú operáciu a blok sa automaticky vráti na pôvodné miesto v paneli (zobrazené animáciou presúvajúceho sa bloku). V tomto prípade blok nie je vložený do schémy.

V aplikácii existujú 2 typy blokov:

- štandardné bloky dostupné v paneli blokov, ktoré sú určené na modelovanie dynamického správania systémov
- „pseudo“ bloky generované automaticky touto aplikáciou, ktoré umožňujú dodatočné používateľské zmeny v schéme. Ide o „node“ blok (uzol) použitý na rozvetvenie prepojení a „line“ blok umožňujúci zmeny polohy úsečiek v prepojení dvoch štandardných blokov

Ako bolo spomenuté vyššie, vloženie bloku do schémy je vykonané v dvoch krokoch. Popri vkladaní vizuálnej reprezentácie bloku (obrázku) do plátna je vytvorená objektová reprezentácia bloku – nová inštancia JavaScriptového objektu reprezentujúceho konkrétny blok. Tento objekt je po vytvorení pripojený na koniec globálneho zoznamu všetkých blokov konkrétnej schémy. Po zistení typu vkladaného bloku (štandardný blok / „pseudo“ blok) je tomuto bloku priradená korešpondujúca trieda CSS. Následne je blok vložený do DOM ako nový XHTML element. Tomuto elementu sú priradené potrebné vlastnosti ako napríklad:

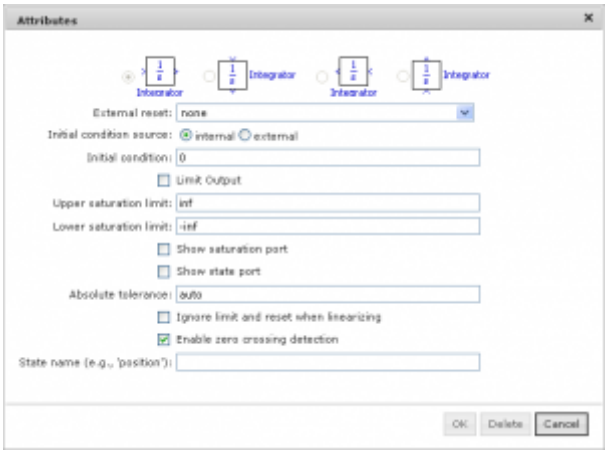
- možnosť presúvania myšou
- funkcie na obsluhu udalostí typu „hover“ a „onClick“ použité na aktivovanie vstupných a výstupných portov bloku
- zobrazenie dialógového okna na zmenu atribútov bloku (iba štandardné bloky) a podobne

To znamená, že hoci sú všetky bloky (vrátane „node“ a „line“ blokov) presúvateľné vnútri hraníc plátna, iba štandardné bloky majú možnosť zmeny atribútov a zvýraznenia (označenia) ich vstupných resp. výstupných portov. Vytvorená inštancia objektu definuje vlastnosti, ktoré zodpovedajú vlastnostiam viditeľného XHTML elementu (pozícia, typ, ...). Okrem týchto vlastností obsahuje objekt aj hodnoty atribútov konkrétneho bloku (množina atribútov je závislá od typu bloku). V konfiguračnom súbore je zoznam atribútov a ich predvolených hodnôt, ktoré sú automaticky nastavené pre každý vytváraný blok.

2.3 Zmeny atribútov jednotlivých blokov

Atribúty bloku môže používateľ zmeniť prostredníctvom dialógového okna zobrazeného po kliknutí na ktorýkoľvek zo štandardných blokov. Toto dialógové okno je štandardný modálny formulár z knižnice jQuery UI, ktorého obsah je načítaný dynamicky prostredníctvom technológie AJAX z iného externého skriptu. Tento skript vytvorí formulár (všetky prvky dialógového okna) z nastavení atribútov zahrnutých v konfiguračnom súbore blocks.xml. Formulár je predvyplnený aktuálnymi hodnotami jednotlivých atribútov zvoleného bloku. V konfiguračnom súbore sú definované aj typy vstupných polí pre atribúty, ktorými môžu byť bežné textové polia, výberové pole (select box), zaškrtávacie pole (checkbox) alebo skupina prepínacích tlačidiel (radio button).

Okrem zmeny atribútov bloku umožňuje toto dialógové okno nastavenie rotácie bloku. Zmena rotácie nie je povolená, ak je blok už pripojený k inému prvku v schéme. Okno s atribútmi poskytuje možnosť vymazania konkrétneho bloku.



Obr. 2. Dialógové okno pre zmenu parametrov integračného člena.

2.4 Prepojenia medzi blokmi

Každý blok má aspoň jeden vstupný alebo výstupný port. Tieto bloky sú graficky vyznačené po presunutí ukazovateľa myši nad daný blok. Na prepojenie dvoch blokov musia byť kliknutím vybraté dva porty. Prepojenie je možné jedine medzi vstupným a výstupným portom dvoch rozdielnych blokov alebo medzi vstupným portom a iným prepojením. V prípade, ak chce používateľ prepojiť blok s iným prepojením, je nutné vybrať najskôr vstupný port bloku a potom kliknúť na požadované miesto v existujúcom spojení.

Na tomto mieste bude automaticky vytvorený nový uzol a vznikne prepojenie medzi týmto uzlom a zvoleným blokom. Vstupné a výstupné porty použité v iných spojeniach sa javia ako neaktívne a nie je ich možné označiť a tým ich zahrnúť do nového prepojenia. Pokiaľ spojenie pozostáva z viac než dvoch úsečiek, do stredu prepojenia je automaticky vložený tzv. „line“ blok. Tento pseudo blok je použiteľný na prispôsobenie pozície strednej úsečky v prepojení.

2.5 Presun blokov

Bloky je možné presúvať v rámci hraníc kresliaceho plátna jednoduchým presúvaním myšou. Po presunutí bloku sú nové súradnice uložené aj do JavaScriptového objektu reprezentujúceho presunutý blok. Následne sú prekreslené všetky prepojenia súvisiace s presunutým blokom. Pseudo bloky „line“ môžu byť presúvané len v priestore medzi dvomi blokmi prepojenými spojením, v ktorom sa nachádza daný „line“ blok.

2.6 Vymazávanie blokov a prepojení

Blok môže byť vymazaný kliknutím na tlačidlo „Delete“ nachádzajúce sa v pravom dolnom rohu okna s atribútmi zvoleného bloku. Spolu s blokom je vymazané aj každé spojenie pripojené k vymazávanému bloku. Pre vymazanie prepojenia stačí kliknúť myšou na požadované prepojenie. Zvolené prepojenie bude graficky zvýraznené, aby sa používateľ mohol presvedčiť o tom, ktoré prepojenie/prepojenia budú v skutočnosti vymazané. V závislosti od miesta, na ktoré používateľ klikol, môžu byť na vymazanie vybraté viaceré prepojenia, aby sa predišlo vytvoreniu neplatných prepojení, napríklad prepojenie dvoch vstupných portov. Vymazanie treba potvrdiť v dialógovom okne, ktoré sa zobrazí po kliknutí na prepojenie.

3. Krátky zoznam najdôležitejších objektov a metód

Najdôležitejšími typmi v tejto aplikácii sú objekty typu „Block“ a „Connection“, čiže bloky a prepojenia.

3.1 Bloky

Blok je najkomplexnejším objektom v celej aplikácii. Špecifikuje svoju pozíciu (horizontálne a vertikálne súradnice bloku v schéme – na plátne), typ (derivačný člen, displej, osciloskop, ...) a rotáciu. Okrem toho inštancia takéhoto objektu uchováva informácie o atribútoch priradených konkrétnemu bloku. Atribúty sú uložené v asociatívnom poli. Vytvorená aplikácia zahŕňa niekoľko metód na jednoduchú manipuláciu s blokovými objektmi. Popísané sú v nasledujúcom zozname. Metódy objektu Block:

- metódy typu getter a setter (pozri slovník pojmov) pre členské premenné: id, type, x, y, version
- špeciálne metódy typu setter pre súradnice, ktoré uvažujú navyše grafické odsadenie blokov (odsadenie je definované v konfiguračnom súbore)
- získanie/nastavenie (get/set) hodnoty atribútu – manipulácia s jediným atribútom (nie s celým poľom)

Doplnkové funkcie:

- getBlockById – vracia referenciu na blok s daným ID
- removeBlock – odstraňuje inštanciu bloku s daným ID

3.2 Prepojenia

Objekt typu Connection musí uchovávať nevyhnutné parametre, ktoré sú potrebné na vykreslenie spojovacích úsečiek medzi dvomi prepojenými blokmi, resp. medzi blokom a uzlom. Spojenie medzi dvomi elementmi je reprezentované ich ID číslami a identifikátormi použitých portov. Blok môže obsahovať viac ako jeden vstupný, resp. výstupný port. Každé spojenie obsahuje aj zoznam identifikátorov úsečiek, ktoré boli použité na vykreslenie prepojenia. Pre objekty spojenia je dostupných takisto niekoľko metód na manipuláciu. Metódy objektu Connection:

- metódy typu getter a setter pre členské premenné: id, startBlockId, startPoint, endBlockId, endPoint, lines

Doplnkové funkcie:

- getConnectionById – vracia referenciu na prepojenie s daným ID
- getConnecionByStartBlockId, getConnectionByEndBlockId – vracajú referencie na prepojenia (ako pole), ktoré spájajú blok s daným ID
- getConnectionByLineId – vracia referenciu na prepojenie, ktoré obsahuje úsečku s daným ID
- removeConnection – odstraňuje inštanciu prepojenia s daným ID

4. Výmena dát s Matlabom

V predchádzajúcej časti bola popísaná práca s kresliacim nástrojom. Hlavným cieľom je však ponúknutie vhodného formátu pre blokové schémy, ktoré môžu byť simulované v prostredí programu Matlab/Simulink. Z toho vyplýva nutnosť exportu vytvorenej blokovej schémy do natívneho formátu Matlabu vo forme súboru .mdl. Ide o textový súbor, ktorý je vytvorený v back-end časti aplikácie na webovom serveri. Súbor je vytvorený na základe údajov z blokovej schémy vytvorenej prostredníctvom predstaveného kresliaceho nástroja. Exportovaný súbor môže byť neskôr použitý na lokálne alebo on-line simulácie.

5. Prehľad technológií

Kresliaci nástroj je vytvorený v značkovacom jazyku XHTML, ktorý bol použitý v kombinácii s kaskádovými štýlmi a skriptovacím jazykom JavaScript. Export a import sú vykonávané skriptami v jazyku PHP prostredníctvom technológie AJAX. Tieto technológie sú zvolené kvôli zachovaniu čo najširšej kompatibility kresliaceho nástroja. Na klientovu stranu nie sú kladené žiadne špeciálne požiadavky. Jedinou vyžadovanou vecou je kompatibilný internetový prehliadač (aplikácia bola testovaná najmä v prehliadačoch FireFox 3.5 a Internet Explorer 8, ale mala by bez obmedzení pracovať v IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome).

Na urýchlenie vývoja a zjednodušenie zdrojových kódov bola použitá JavaScriptová knižnica jQuery, ktorá sa využíva na manipuláciu s objektmi, zmeny CSS atribútov, zobrazovacie funkcie a komunikáciu prostredníctvom AJAX požiadaviek. Na strane servera by mala byť pre PHP nainštalovaná minimálne verzia 5.2 alebo vyššia. Verzia PHP 5.2 má implementovanú podporu objektov JSON, ktoré sú využívané na výmenu dát medzi front-endovou aplikáciou a skriptami realizujúcimi export/import blokových schém.

6. Slovník pojmov

- DOM (document object model) – reprezentácia obsahu XHTML dokumentu v stromovej štruktúre
- element – všeobecné pomenovanie pre bloky, uzly alebo prepojenia
- getter – jednoduchá metóda (funkcia) používaná na získanie hodnoty požadovanej členskej premennej
- inštancia objektu – reprezentácia konkrétneho objektu v aplikácii (napríklad blok s id = „b_0“, vložený na pozícii so súradnicami [23,40], type = „step“, version = 1)
- objekt – reprezentácia nejakej entity (napr. blok – Block, prepojenie – Connection, pozícia – Position, …)
- setter – jednoduchá metóda (funkcia) používaná na nastavovanie hodnoty požadovanej členskej premennej
- XHTML element – časť XHTML dokumentu (podobne ako „tag“ v dokumente HTML) – uzol DOM, napríklad , <img>, …

7. Záver

Popísaná aplikácia nájde svoje uplatnenie pri riadení dynamických systémov v rámci virtuálnych a vzdialených laboratórií. Vzhľadom na to, že je vyvinutá najrozšírenejšími technológiami, môže byť kombinovaná s väčšinou riešení umožňujúcich použitie

Matlabu cez Internet. Aplikácia by mala priniesť pohodlný a intuitívny nástroj rovnako pre študentov ako aj pre pedagógov, resp. iných záujemcov z radov verejnosti pracujúcej s programovým prostredím Matlabu.

8. Odkazy na literatúru

1. S. Holzner, Mistrovství v AJAXu, Computer Press. Brno, 2007.
2. J. Resig and the jQuery Team. 2009. jQuery Documentation. http://docs.jquery.com
3. Chr. Schmid, „Internet – basiertes Lernen“, Automatisierungstechnik, 51, No. 11, p. 485-493, 2003.
4. K. Žáková, M. Sedlák, “Remote Control of Experiments via Matlab”, Int. Journal of Online Engineering (iJOE), 2, No. 3, 2006.
5. Refsnes Data. 1999-2009. JavaScript and HTML DOM Reference. http://www.w3schools.com/jsref/default.asp
6. The PHP Group. 2001-2009. PHP Documentation. http://www.php.net/docs.php

---

Spoluautorkou článku je doc. Ing. Katarína Žáková, PhD., Ústav riadenia a priemyselnej informatiky, Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave