



# Matlab (5)

## Stateflow – modelovanie udalostných systémov

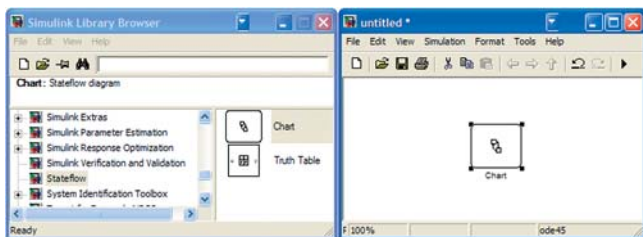
V druhej časti nášho seriálu o Matlabe a produktoch s ním prepojených sme sa oboznámili s graficky orientovaným prostredím na návrh dynamických modelov Simulink. Simulink predstavuje nástroj vhodný na tvorbu modelov, pri ktorých nás zaujíma ich správanie v čase. Pribehy jednotlivých veličín sú závislé od vstupov a od času. V tejto časti by sme vám radi predstavili produkt, ktorý sa sústreďuje na sledovanie udalostí. Ide o produkt Stateflow.

Prímerne je určený na modelovanie udalostných systémov, teda systémov, ktoré reagujú na určitú udalosť, prípadne sekvenciu udalostí. Jedinečnosť programu Stateflow je skrytá v jeho plnej integrácii do výpočtového prostredia Matlab/Simulink. Podobne ako ostatné produkty aj Stateflow je distribuovaný pre rôzne operačné systémy, ako MS Windows, Solaris, Mac OS X alebo Linux.

Stateflow je interaktívny graficko-vývojový nástroj spolupracujúci so Simulinkom na modelovanie a simulovanie udalostne riadených systémov. Udalostne riadené systémy prechádzajú z jedného operačného módu do ďalšieho na základe reakcie na udalosti a podmienky. Udalostne riadené systémy sú zvyčajne modelované na základe teórie konečných automatov. Tie reprezentujú operačné módy ako stavy. Na konštrukciu konečných automatov Stateflow poskytuje grafické objekty, ktoré stačí jednoducho uchytíť myšou a premiestniť na požadované miesto a sériu prechodov na označenie smeru prechodov z jedného stavu do ďalšieho. Keďže Stateflow reprezentuje udalostne riadený systém v rámci Simulinku, môžeme ho pridať do schémy zo Simulink Library Browser alebo pomocou príkazu

```
>>sfnew
```

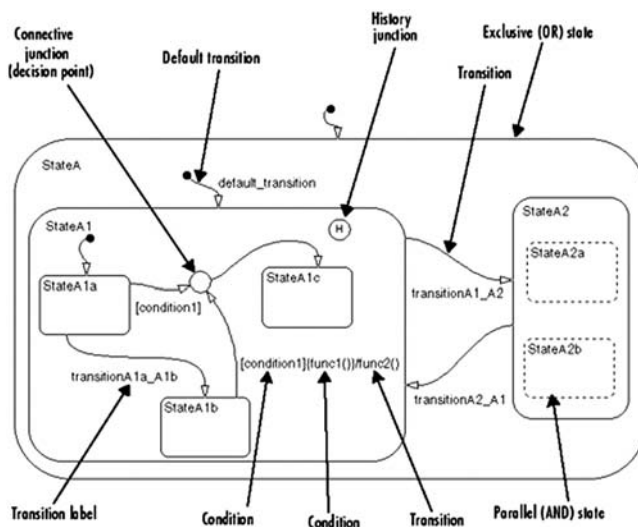
z Matlab Command Window, ktorý vytvorí schému Simulinku (obr. 1 vľavo).



Obr.1 Stateflow v prostredí Simulink

Dvojklikom na blok Chart sa otvorí Stateflow editor, kde môžeme pomocou objektov vytvoriť udalostne riadený systém. Objekty Stateflow delíme na grafické a negrafické. Niektoré grafické objekty vidíme na obr. 2. Patria k nim stavy, prechody, štandardné prechody, história, uzly, pravdivostné tabuľky, grafické funkcie a vnorené funkcie Matlabu. Grafické objekty môžeme zakresliť priamo do grafu. Medzi negrafické objekty patria udalosti, dáta a koncové objekty, ktoré majú iba textovú reprezentáciu.

Základným prvkom konečného automatu sú stavy. Stavy môžu byť aktívne alebo neaktívne. Ak je stav aktívny, systém sa nachádza v tomto móde a naopak, ak je stav neaktívny, systém v tomto móde nie je. Stavy môžu mať vlastnú hierarchiu. Najvyšším stavom je pochopiteľne samostatný graf (Chart). Preto ho nazývame rodičovským stavom, pričom ďalšie stavy sú jeho potomkovia. Podobné vzťahy môžeme definovať na ktorejkoľvek úrovni. Každý stav má svoju dekompozíciu, ktorá hovorí, aký typ podstavov stav obsahuje. Prvý typ dekompozície je exkluzívny stav (XOR, obr. 2 – stav StateA1). V tomto type dekompozície môže byť aktívny iba jeden podstav. Druhý typ dekompozície



Obr.2 Grafické objekty Stateflow

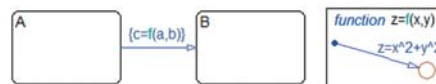
je paralelný stav (AND, obr. 2 – stav StateA2). V tomto type dekompozície môžu byť aktívne všetky podstavy naraz. Každému stavu môžeme definovať nasledujúce vlastnosti:

**name/**  
**entry:entry actions**  
**during:during actions**  
**exit:exit actions**  
**bind:events, data**  
**on event\_name:on event\_name actions**

Prvou vlastnosťou je meno stavu. Ďalšie tri vlastnosti nám hovoria, čo sa má vykonať pri vstupe do stavu (entry), počas aktivity stavu (during) a pri opúšťaní stavu (exit). Ďalšia vlastnosť (bind) zabezpečuje, že sa s dátami môže manipulovať len v rámci aktívneho stavu, pričom ostatné stavy môžu danú hodnotu len používať, nie však nemeniť. Posledná vlastnosť určuje reakciu na akúkoľvek udalosť v systéme. Jednotlivé vlastnosti stavu môže, ale aj nemusí obsahovať. Prechod je čiara, ktorá spája objekty a väčšinou ide o prechod medzi stavmi. Prechody obsahujú podobne ako stavy vlastnosti.

**event[condition]{condition\_action}/transition\_action**

Prvá vlastnosť (event) hovorí, že sa má prechod spustiť, ak nastane daná udalosť. Druhá vlastnosť (condition) definuje podmienku, ktorá musí byť splnená, aby mohol prechod nastať. Posledné vlastnosti definujú, čo sa má vykonať, keď je splnená podmienka prechodu (condition\_action), alebo keď nastane prechod (transition\_action).



Obr.3 Grafická funkcia



Ako sme už spomínali skôr, stav v exkluzívnej dekompozícii môže mať aktívny iba jeden podstav. Vynára sa otázka, ako Stateflow vie, do ktorého stavu má v tomto prípade vstúpiť. Zabezpečíme to objektom štandardný prechod (default transition). Na obr. 2 vidíme, že máme v stave StateA na výber dva stavy, ale štandardný prechod nám hovorí, že ako do prvého sa pôjde do stavu StateA1. Na vetvenie prechodov slúži objekt uzol (connective junction). Pomocou tohto objektu môžeme vytvárať konštrukcie ako rozhodovanie typu if-then, cykly for, vetvenie prechodu z jedného zdroja na viacero cieľov a spájanie z viacerých zdrojov do jedného cieľa. Objekt história (history junction) slúži na zapamätanie si posledných podstavov stavu, ktorý obsahuje históriu. Pri následnom vrátení sa do tohto stavu sa aktivuje posledný aktívny stav. Grafická funkcia je nástroj, ktorý poskytuje grafické rozhranie na definovanie štandardných funkcií. Príklad grafickej funkcie je na obr. 3. Samozrejme môžeme definovať zložitejšie funkcie, v ktorých vieme vytvárať cykly a rozhodovaciu logiku. Veľká výhoda grafických funkcií je možnosť sledovania tokov postupností operácií priamo v grafe počas simulácie.

Ďalšou vlastnosťou Stateflow je, že umožňuje využitie vnorených funkcií priamo z Matlabu. Jednoducho stačí použiť objekt Embedded Matlab function, do ktorého sa iba napíše meno funkcie a dvojklikom na objekt môžeme našu funkciu editovať.

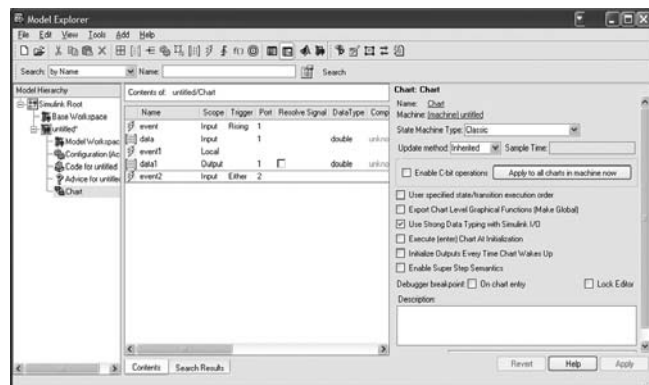
V prvej časti sme sa venovali grafickým objektom, teraz sa pozrieme, ako sa pracuje s udalosťami a dátami. Stateflow rozpoznáva dve udalosti a takzvané volanie funkcie [2]. Z dvoch udalostí môžeme teda reagovať na nábehovú (rising trigger), dobehovú (falling trigger) a obe hrany (either trigger). Udalosti vieme definovať ako vstup zo Simulinku, lokálne a výstup zo Simulinku. Pridávanie udalosti môžeme robiť v menu takto:

**Add – event – Local**

**Add – event – Input from Simulink**

**Add – event – Output to Simulink**

Vynikajúcou pomôckou na pridávanie udalostí je Model Explorer (obr. 4). Samozrejme, že neslúži iba na pridávanie udalosti. Môžeme si v ňom pozrieť celú štruktúru Chartu, definovať udalosti, dáta a meniť všetky vlastnosti.



Obr.4 Stateflow Explorer

Stateflow podobne ako udalosti dovoľuje definovať aj dáta. Takisto ich môžeme získať priamo zo schémy Simulink (na vstupe alebo na výstupe), definovať ich ako lokálne premenné alebo ako konštanty. Využíваме pritom štandardné dátové typy Matlabu, ktoré sa zobrazia aj v menu. Pridať dáta môžeme pomocou menu alebo pomocou už spomínaného Stateflow Explorera.

**Add – data – Local**

**Add – data – Input from Simulink**

**Add – data – Output to Simulink**

**Add – data – Constant**

Stateflow je výborným nástrojom na modelovanie konečných automatov. Je veľmi prehľadný a intuitívny, priam až jednoduchý na používanie. Človek oboznámený s teóriou konečných automatov si veľmi rýchlo zvykne na používanie tohto nástroja z dielne The MathWorks. Nakoľ-

ko ide o jeden z mála produktov, ktorý sa dokáže začleniť do spojitých modelov a doplniť ich o udalostné systémy, stal sa vo svete populárnym. Jeho služby využívajú popredné spoločnosti venujúce sa návrhu a výskumu v oblasti riadenia a modelovania.

Článok vznikol s podporou APVV-99-045805.

## Literatúra

[1] Stateflow 7, Getting Started Guide, The MathWorks

[2] Stateflow® and Stateflow® Coder™ 7 User's Guide, The MathWorks

[3] <http://www.mathworks.com/products/stateflow/>

[4] <http://www.mathworks.com/access/helpdesk/help/toolbox/stateflow/>

**Ing. Martin Foltin, PhD.**

Slovenská technická univerzita v Bratislave  
Fakulta elektrotechniky a informatiky  
Ústav riadenia a priemyselnej informatiky  
Ilkovičova 3, 812 19 Bratislava  
e-mail: martin.foltin@syprin.sk

**Ing. Michal Blaho**

Slovenská technická univerzita v Bratislave  
Fakulta elektrotechniky a informatiky  
Ústav riadenia a priemyselnej informatiky  
Ilkovičova 3, 812 19 Bratislava  
e-mail: michal.blaho@syprin.sk

46