

CHEMICKO-INŽENÝRSKÝ OBJEKTOVÝ MODEL

Svět a zejména Evropa se v rámci globalizace standardizuje. Normuje se kde co a důležité to je zejména v oblasti informatiky a výpočetní techniky, kde to usnadňuje další bouřlivý rozvoj. Proč by chemické inženýrství mělo stát stranou? Řada rysů a postupů je totiž obecná a společná všem chemicko-inženýrským výpočtům, tak proč je nezařadit do standardizovaného objektového modelu, který se osvědčuje u programovacích jazyků? Uživatel se jimi pak už nemusí do detailu zabývat, pouze je deklaruje z knihoven a vytvoří své vlastní instance. Myslím si, že takový objektově orientovaný model by měl být vytvářen zdola na základě praktických potřeb nejlépe jako Open Source nebo akademická iniciativa, aby nepodléhal komercializaci. Mohlo by se to stát tématem diplomových prací. Věřím, že se toho nějaká univerzita ujme. Takový model pak může sloužit při výpočtech a modelování v různých moderních programovacích jazycích C++, C#, Java nebo Visual Basic.

Existuje řada komerčních univerzálních simulačních programů jako Aspen [1], ChemCAD [2], PRO/II [3], Fluent [4] a další. Umožňují složité a komplexní simulace, ale mají také své nevýhody. Vzhledem k jejich ceně ve statisících korun jsou dostupné jen malému počtu inženýrů a jejich proprietární struktura nebývá příliš dokumentovaná. Jejich složitá obsluha představuje další dovednost. Navíc o velké simulační programy tu nejde. Většina chemických inženýrů provádí a programuje své jednoduché nebo střední výpočty sama v rámci studia nebo vývoje. Řadu opakovaných úkonů spojených s definováním složek, směsí, parametrů proudů musí vždy provádět a definovat znovu. Je to netvůří rutinní práce. A tady by právě mohl významně pomoci základní objektový model. Zprvu by zahrnoval jen ty základní jednoduché atributy chemických výpočtů jako definovat složky, vlastnosti, směs, proud. Ale uživatel by se o ně už nemusel starat a pouze by k svému programu přilinkoval veřejně dostupnou dynamickou knihovnu .dll s těmito objekty. Tak jako programátor deklaruje použití celočíselné nebo textové proměnné, inženýr bude ve svém programu pouze deklarovat použití směsi, složek, proudů.

V následujícím odstavci jsem se pokusil ideově nastínit, jak by zárodek takového objektového modelu mohl vypadat. Tučná kurzíva označuje *Objekt* a *()* pak pole, jak obvyčejné, tak pole objektů. Objekt *Latka* představuje chemické individuum nebo charakteristickou složku směsi, popisovanou jak bodovými, tak teplotně případně tlakově závislými údaji. Tyto závislé vlastnosti je dobré vyjádřit formou objektu *Vlast* spolu se zapouzdřenými koeficienty korelačních vztahů načítaných z databáze.

Pole nebo kolekce látek pak vytváří základ vyššího objektu *Smes*, který už také nese údaj o složení. Chemický inženýr dále pracuje s proudy, které, kromě složení, mají i určitou teplotu a tlak. Jsou topologicky orientovány, a proto si zaslouží samostatný objekt *Proud*. Na základě těchto základních (a možná ještě dalších) objektů je možné vytvářet další objekty vázané k určitému typu aparátů nebo operací.

Latka název, číslo, sumární vzorec, CAS, pole bodových vlastností, pole vlastností *Vlast()*

Vlast název, konstanty teplotního rozvoje, typ korelačního vztahu, meze platnosti, třída přesnosti

Smes název směsi, počet složek, pole s čísly složek, *xhmot()*, *xmol()*, **Latka()** pole se složkami směsi

Proud název, číslo, hm. průtok, T,P, *xhmot()*, *xmol()*, č.apartátu z kterého proud vychází, č.apartátu do kterého proud vstupuje, *Smes* - objekt s údaji o složkách směsi

Vymeník název, číslo, **Proud_trubky**, **Proud_plast**, parametry trubkovnice, pláště výměníku, metody pak jako koef. prostupu, přestupu, převedené teplo, tlaková ztráta ...

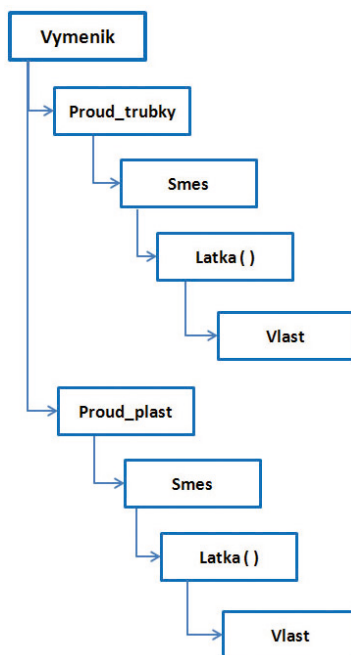
obdobně další aparáty

Reaktor

Kolona

Mixer...

Obr. – Příklad hierarchie objektu Vymeník



Z hlediska metod objektů, které představují vlastní funkčnost, to budou jednak servisní metody načtení a validace dat (set), jejich zobrazení (get) a jednak vlastní zabezpečení funkcí. U objektu *Smes* výpočet jednotlivých fyzikálních vlastností směsi

pro zadané složení (*xmol()*), teplotu a tlak. U objektu *Vymeník* výpočet koef. přestupu a prostupu, iterační výpočet výměníku, tlakových ztrát apod. Výhodné bude samozřejmě využívat i nějakého univerzálního modelu matematických numerických metod.

Slyším vaši námitku, že obecný model nemůže vyhovovat všem, protože každý si potřebuje svůj výpočet specificky upravit vzhledem k svým potřebám. Ale i na to objektový přístup pamatuje formou dědičnosti a přetížením metod. Nevyhovuje-li vám třeba standardní objekt proudů nebo výměníku, vytvoříte si vlastní odvozený, který se postará o většinu rutiny, a vy si pouze změňte nebo přidáte potřebné parametry a změňte (tzv. přetížíte) nevhodné metody výpočtu. To umožňuje, aby byl programový modul zároveň uzavřený a zároveň otevřený. Uzavřený, že jeho uživatel nemusí ba někdy ani nemůže nic přidávat do rutinních částí, kam by spíše mohl vnést náhodnou chybu, ale současně má možnost některé pro něj nevhodné části změnit a upravit podle své potřeby. Dědičnost objektů nám umožňuje:

- to, co bylo v základním objektu dobré bez dalšího úsilí ponechat,
- to, co nám tam chybělo jednoduše dodat,
- to, co se nám tam nelíbilo změnit.

Typickým příkladem je třeba kondenzátor. Lze ho odvodit ze základního objektu výměníku tak, že zdědí většinu atributů, ale vytvoří se pro něj některé nové proměnné a charakteristické metody výpočtu přestupu při kondenzaci, podchlazení kondenzátu, problematika inertů a podobně. Obdobně v případě reaktorů, kolon i dalších aparátů. Vytvoření objektově orientovaného modelu pro oblast chemického inženýrství není tedy ani tak o vztazích jako o hierarchické struktuře. Zpočátku by byl jen malý a jednoduchý, ale mohl by se dále rozrůstat. Konec konců k Matlabu také existuje bezplatná alternativa Octave [5]. Existence takovéto volně dostupné knihovny by jistě usnadnila a zefektivnila výpočty. Cílem tohoto článku je vzbudit zájem a vyvolat případnou diskusi.

Literatura

- [1] <http://www.aspentech.com/core/aspentech-plus.cfm>
- [2] <http://www.chemstations.net/>
- [3] <http://ips.invensys.com/en/products/processdesign/Pages/ProcessingEngineeringSuite-P0021.aspx>
- [4] <http://www.fluent.com/news/pr/pr1.htm>
- [5] <http://www.gnu.org/software/octave/>

Vratislav HLUBUČEK, Neratovice,
vratislav.hlubucek@centrum.cz