

# VYBRANÉ KLIENTSKE TECHNOLOGIE PRI MODELOVANÍ PROCESOV S VYUŽITÍM WEBOVEJ SLUŽBY

HOROVČÁK P., TERPÁK J., CIRBES P., KUKURUGYA J.

Technická univerzita v Košiciach, Fakulta BERG, Ústav riadenia a informatizácie výrobných procesov, Košice, pavel.horovcak@tuke.sk

*Lubovoľnú výrobu, resp. technologickú linku je možné rozdeliť na určitý počet typových členov, v ktorých prebiehajú elementárne procesy, napr. difúzia, výmena tepla, chemická reakcia a iné. Proces predstavuje vnútorné vzťahy v systéme, t.j. spôsob transformácie vstupov do systému na výstupy. Iným spôsobom povedané ide o transformáciu substancie. Podľa typu transformácie rozlišujeme tri základné skupiny procesov ako aj ich zodpovedajúce modely, a to modely procesov prenosu (substancia mení svoju polohu), modely procesov akumulácie (hromadenie substancie) a modely procesov premeny (substancia sa mení na inú).*

Substancia je ľubovoľná realita, ktorá je predmetom transformácie. Môže to byť voda, plyn, elektrická energia, ruda, železo, oceľ alebo aj informácia. Procesy transformácie informácií voláme aj kybernetické procesy. Vlastnosti substancie môžeme rozdeliť na vlastnosti, ktoré sú závislé na jej množstve – extenzívne vlastnosti (napr. hmotnosť, objem, látkové množstvo) a vlastnosti nezávislé na množstve – intenzívne vlastnosti (napr. hustota, teplota, viskozita).

V prípade procesov prenosu dochádza k transformácii polohy substancie. Vo všeobecnosti môžeme procesy prenosu rozdeliť na procesy prenosu hybnosti, hmoty a energie.

Ustálené procesy prenosu hybnosti, hmoty a tepelnej energie sú popísané podobnými rovnicami ako napr. Ohmov zákon pri toku elektrického prúdu. Konkrétne rovnice pre jednotlivé procesy sú :

$$\text{proces prenosu hybnosti } i_p = -v \frac{\rho dw}{dx}, \left( \frac{\text{kg} \cdot \text{m} \cdot \text{s}^{-1}}{\text{m}^2 \cdot \text{s}} \right) \quad (1)$$

$$\text{proces prenosu hmoty difúziou } i_n = -D \frac{dc}{dx}, \left( \frac{\text{mol}}{\text{m}^2 \cdot \text{s}} \right) \quad (2)$$

$$\text{a proces prenosu tepelnej energie } i_Q = -a \frac{\rho c dt}{dx}, \left( \frac{\text{J}}{\text{m}^2 \cdot \text{s}} \right) \quad (3)$$

Zákonitosti všetkých troch procesov môžeme preto zovšeobecniť nasledovne. Tok substancie (intenzita hybnosti, hmoty a energie) je priamo úmerný hnacej sile (gradientu rýchlosti, koncentrácie a teploty) a nepriamo úmerný odporu, t.j.

$$\text{intenzita} = \frac{\text{hnacia sila}}{\text{odpor}}, \text{ resp. intenzita} = \frac{\text{rozdiel potenciálov}}{\text{odpor}} \quad (4)$$

Odpor pre jednotlivé procesy je priamo úmerný vzdialenosti  $dx$  a nepriamo úmerný dynamickej viskozite  $\mu = v \cdot \rho$  ( $\text{kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$ ), difúznemu koeficientu  $D$  ( $\text{m}^2 \cdot \text{s}^{-1}$ ) a tepelnej vodivosti  $\lambda = a \cdot \rho \cdot c$  ( $\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$ ).

Taktiež porovnanie rovníc trojrozmerného neustáleného procesu

$$\text{prenosu hybnosti } \frac{\partial w}{\partial \tau} = v \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \quad (5)$$

$$\text{prenosu hmoty difúziou } \frac{\partial c}{\partial \tau} = D \left( \frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2} \right) \quad (6)$$

$$\text{a prenosu tepla vedením } \frac{\partial t}{\partial \tau} = a \left( \frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} + \frac{\partial^2 t}{\partial z^2} \right) \quad (7)$$

ukazuje, že všetky uvedené procesy sú analogické.

Analogické sú aj koeficienty úmernosti jednotlivých procesov a to kinematická viskozita  $v$ , difúzny koeficient  $D$  a teplotná vodivosť  $a$ . Rozmer všetkých týchto fyzikálnych veličín je rovnaký  $\text{m}^2 \cdot \text{s}^{-1}$ .

Anológia rovníc umožňuje taktiež použitie rovnakých metód pre ich riešenie. Analytické riešenie je u väčšiny praktických úloh veľmi náročné, takže sa najčastejšie využívajú numerické metódy.

Podobne ako v prípade procesov prenosu aj akumuláciu môžeme zovšeobecniť a popísať ako prírastok príslušnej hodnoty hybnosti,

hmoty a energie za čas  $d\tau$ , ktorý je ekvivalentný súčinu príslušnej kapacity a prírastku rýchlosti, koncentrácie, resp. teploty za čas  $d\tau$ , t.j.

$$\text{akumulácia hybnosti } \frac{dp}{d\tau} = m \frac{dw}{d\tau} \quad (8)$$

$$\text{akumulácia hmoty } \frac{dn}{d\tau} = V \frac{dc}{d\tau} \quad (9)$$

$$\text{a akumulácia tepelnej energie } \frac{dQ}{d\tau} = C \frac{dT}{d\tau} \quad (10)$$

kde  $p$  je hybnosť ( $\text{kg} \cdot \text{m} \cdot \text{s}^{-1}$ ),  $n$  – látkové množstvo (mol),  $Q$  – množstvo tepelnej energie (J),  $w$  – rýchlosť ( $\text{m} \cdot \text{s}^{-1}$ ),  $c$  – koncentrácia ( $\text{mol} \cdot \text{m}^{-3}$ ),  $T$  – teplota (K). Kapacity pre jednotlivé prípady sú hmotnosť  $m$  (kg) – akumulácia hybnosti, objem  $V$  ( $\text{m}^3$ ) – akumulácia hmoty a tepelná kapacita  $C$  ( $\text{J} \cdot \text{K}^{-1}$ ) – akumulácia tepelnej energie.

Modely procesov premeny predstavujú procesy kde sa jedna substancia mení na inú. Typickými príkladmi procesov premeny sú procesy spaľovania palív, procesy rozkladu uhlíkatých, oxidácia, resp. redukcia oxidov, tvorba okovín a pod. Uvedené procesy premeny sú súčasťou technologických procesov spracovania surovín. Procesy premeny je možné popísať z troch hľadísk, t.j. z hľadiska dynamiky, kinetiky a statiky.

V procesoch premeny sa stretávame s veľkým počtom chemických reakcií a fázových premen. Dynamika predstavuje hodnotenie procesu, resp. chemickej reakcie, z hľadiska pôsobenia síl, t.j. či reakcia môže prebiehať v naznačenom smere. Gibsová voľná energia je vhodným a použiteľným kritériom na posúdenie hnacej sily chemickej reakcie.

Okrem smeru chemickej reakcie je pre proces premeny rozhodujúca rýchlosť priebehu daného procesu. Pod rýchlosťou chemickej reakcie budeme uvažovať zmenu látkového množstva reagujúcich zložiek za jednotku času, vyvolanú ich vzájomným pôsobením v jednotke reakčného objemu (homogénne reakcie), alebo na jednotke povrchu, resp. v jednotke hmotnosti (heterogénne reakcie).

Priebeh niektorých procesov premeny je v porovnaní s ostatnými procesmi pomerne rýchly, a preto ho môžeme modelovať z hľadiska statiky, t.j. zohľadňujeme stav pred a po úplnom priebehu procesu premeny.

Výsledný model technologického procesu potom predstavuje syntézu elementárnych procesov a vo všeobecnosti vychádza zo zákonov zachovania, t.j. je daný bilanciou hybnosti, hmoty a energie a má nasledujúci tvar

$$\text{akumulácia} = \text{prenos} + \text{premena},$$

kde akumulácia je vyvolaná procesmi prenosu a premeny hybnosti, hmoty a energie v danom bilančnom objeme [1–6].

Z popisu modelov elementárnych procesov jednoznačne vyplýva potreba jednotného a ľahko dostupného zdroja termodynamických, tepelno-fyzikálnych a fyzikálnych vlastností substancii. Predložený článok predstavuje jednu z možností ako takýto jednotný zdroj vo forme webovej služby používať.

## Technológie klientov

Základnou požiadavkou pre výstavbu klientskej aplikácie, ktorá využíva pre svoju činnosť webovú službu, je znalosť referenčného súboru služby, tzv. WSDL (Web Service Description Language) súboru služby. Tento súbor obsahuje všetky potrebné špecifikácie danej webovej služby. Patria tam aj názvy jednotlivých funkcií webovej služby spolu so zoznamom a typmi jednotlivých vstupných, prípadne výstupných parametrov týchto funkcií. Súčasne sa tam nachádza aj URI (Unified Resource Locator) danej webovej služby, ktorý je neskôr využitý pre vytvorenie priameho pripojenia klientskej aplikácie k webovej službe a ich vzájomnej komunikácie. Tvorbu klientskej aplikácie s využitím webovej služby podporujú všetky aktuálne vývojové prostredia. Predovšetkým sú to internetové vývojové prostredia, ale rovnako aj desktopové vývojové prostredia. V ďalšom uvedieme vytvorenie klienta v štyroch najrozšírenejších prostrediach. Pre možnosť vzájomného porovnania jednotlivých technológií bola zvolená jednoduchá klientska aplikácia, spočívajúca vo výpočte zadanej skupiny termodynamických hodnôt zvolenej látky.

### 1.1 Klient PHP

PHP (PHP Hypertext Processor) je veľmi rozšírený a všestranne použiteľný skriptovací jazyk, ktorý je určený špeciálne pre vývoj webu, resp. webových aplikácií. Bol vytvorený pre prácu na webe a môže byť vložený do HTML. Je to procedurálny jazyk, v aktuálnej verzii 5 zaznamenal výrazný prechod na objektovo orientované vybavenie.

Realizácia klientskej aplikácie je založená na vytvorení inštancie objektu, ktorý reprezentuje klientsku časť webovej služby. Druhým krokom je volanie potrebnej funkcie webovej služby s príslušnými hodnotami vstupných parametrov funkcie. Následne je potrebné zobraziť výstupné hodnoty webovej služby. Vytvorenie inštancie objektu má tvar:

```
$urlwsdl='http://omega.tuke.sk/pavel.horovcak/php_ws/wsd/termprop.wsdl';
```

```
$SluzbaTemp = new SoapClient ($urlwsdl);
```

kde `$urlwsdl` udáva umiestnenie WSDL súboru webovej služby (jej URI) a `$SluzbaTemp` predstavuje inštanciu objektu pre komunikáciu s danou webovou službou. Pretože webová služba vracia ako výstupnú hodnotu XML štruktúru, najjednoduchším spôsobom jej prezentácie je priamy výpis tejto štruktúry, ktorý môže mať tvar

```
print $SluzbaTemp->getTermProp($rez,$id,$t);
```

príčom `getTermProp()` je názov funkcie webovej služby. Uvedená funkcia má tri parametre s významom `$rez` – režim práce funkcie, `$id` – identifikácia zadanej látky a `$t` – teplota. Vzhľad prezentácie výsledkov na obrazovke je vhodné zabezpečiť pomocou CSS súboru, v ktorom je pre každý konkrétny element výstupnej XML štruktúry webovej služby predpísaný príslušný spôsob zobrazenia (predovšetkým vhodnou farebnosťou jednotlivých položiek). Položky XML štruktúry, ktoré je potrebné z akýchkoľvek dôvodov nezobrazovať, je možné označiť voľbou `display: none`.

**Požiadavky.** Pre tvorbu klienta v prostredí PHP uvedeným spôsobom je potrebná verzia PHP 5 a vyššie. V aktuálnych distribúciách Linuxu je táto verzia už obsiahnutá. Teda na prevažnej väčšine webových serverových riešení je možné počítať s jej existenciou a preto je tvorba klienta veľmi jednoduchá. Pokiaľ ide o prehliadače, všetky bežné a rozšírené prehliadače (Internet Explorer, Firefox, Opera atď.) sú úplne vyhovujúce.

**Výhody.** PHP je licencované ako open source, je teda dostupné každému záujemcovi. Je to moderný, vysokovýkonný a najrozšírenejší vývojový prostriedok s rozsiahlou podporou všetkých potrebných funkcionalít.

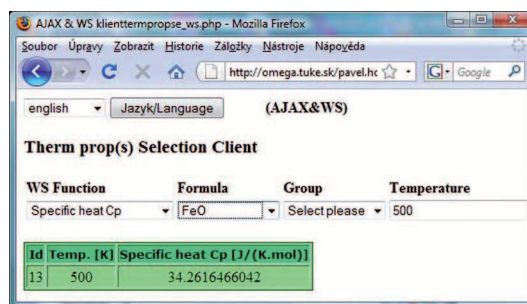
**Nevýhody.** Medzi nevýhody je možné zaradiť náročnejší (napr. v porovnaní s ASP) vývoj a ladenie aplikácie, najmä v prípade rozsiahlych a komplexných projektov.

## 2.2 Klient Ajax

AJAX je skratkou slov Asynchronous Javascript And XML. Je to webová vývojová technológia (presnejšie skupina technológií – XHTML, CSS, DOM, XML, XSLT, XMLHttpRequest a JavaScript) určená na vytváranie interaktívnych webových aplikácií. Predstavuje ďalší logický krok vo vývoji SOA (Service Oriented Architecture). Doslova sa vrátil na scénu webových aplikácií a priniesol vysoko interaktívne možnosti veľmi podobné desktopovým, lebo mení obsah stránok bez nutnosti ich opätovného načítania [7]. Pomocou Ajaxu môžu používateľské rozhrania cez prehliadač využívať webové služby ako svoj dátový zdroj na ukladanie a obnovovanie informácií.

**Realizácia** klientskej aplikácie je založená na vytvorení troch vzájomne spolupracujúcich súborov HTML, JavaScript a PHP. HTML súbor slúži na vstup údajov a požiadaviek používateľa a zobrazenie výstupných hodnôt. JavaScript súbor vytvára podľa typu prehliadača príslušný objekt XMLHttpRequest [8] a na základe vstupných hodnôt z formulára HTML z nich vytvorí dotazovací reťazec a prostredníctvom objektu XMLHttpRequest nadviaže spojenie so súborom PHP, označovaným často ako brána. Tento súbor realizuje vlastné volanie webovej služby a jej výsledok odosiela späť JavaScript súboru. Tento po úspešnom získaní odpovede (objektom XMLHttpRequest) vytvára z prijatej XML štruktúry zodpovedajúce HTML elementy s ich obsahmi, ktoré potom vhodne vypublikuje do príslušného pomenovaného elementu (funkciou `innerHTML`) súboru HTML. Prenos údajov zo servera môže byť orientovaný do jedného alebo viacerých HTML elementov. Podobne jeden objekt XMLHttpRequest môže obstarávať rôzne režimy komunikácie, alebo každej komunikácii môže byť priradený samostatný objekt XMLHttpRequest, prípadne môže byť využívané celé pole týchto objektov. To všetko závisí na rozložení vstupných prvkov stránky, na koncepcii ich funkcionality a na koncepcii asynchrónnej komunikácie medzi klientom a serverom (obr. 1). V našom prípade komunikáciu zabezpečuje jeden XMLHttpRequest objekt, ktorý odosiela na súbor PHP tri parametre a prijíma jednu alebo viac n-tíc údajov (vypočítaných alebo z príslušnej databázovej tabuľky či skupiny tabuliek) dodaných webovou službou. Podstatná časť činnosti klienta je teda sústredená v súbore JavaScript.

Obr. 1 – Ilustrácia Ajax klienta webovej služby



**Požiadavky.** Pre tvorbu klientskej aplikácie v časti HTML a JavaScript nie sú žiadne požiadavky, pre časť PHP je potrebná verzia PHP 5 a vyššie. Vytváranie pomerne veľmi rozsiahlej časti JavaScript štandardným spôsobom je náročné, preto je vhodné vlastnoručne programovanie kódu JavaScript zjednodušiť pomocou niektorého vhodného pracovného rámca pre Ajax, tzv. frameworku. Ide v podstate o knižnicu v JavaScripte, ktorú je možné pripojiť a využiť pre svoje stránky. V súčasnosti existujú desiatky voľne dostupných frameworkov pre všetky hlavné serverové technológie – PHP, ASP.NET a JSP [9]. Urýchľujú vývoj aplikácie tým, že už obsahujú naprogramovanú nízkoúrovňovú komunikáciu medzi klientom a serverom. Po pripojení frameworku stačí potom iba volať príslušné funkcie knižnice [10]. Medzi známe frameworky patrí napr. JPSpan, DWR, Majac, Sack, XHConn, AjaxRequest, Sarissa (pre prácu s XML) atď.

Dokončení na ďalšej strane

**Výhody.** Tak ako PHP aj JavaScript a súvisiace knižnice (frameworky) sú licencované ako open source a teda sú všeobecne dostupné. Po pochopení základných princípov a funkcionality Ajaxu je možné dosiahnuť efektívnu tvorbu moderných a používateľsky optimálnych (user friendly) aplikácií, využívajúcich webové služby.

**Nevýhody.** Pretože každá aplikácia v Ajaxe je tvorená skupinou spolupracujúcich súborov, pričom JavaScript súbor býva veľmi rozsiahly, ich vlastnoručná príprava a najmä ladenie predstavuje oveľa náročnejšiu úlohu v porovnaní s tvorbou obyčajných stránok.

### 2.3 Klient ASP

ASP (Active Server Pages), označované tiež ako klasické ASP, je skriptovacia platforma spoločnosti Microsoft, primárne určená na dynamické spracovanie webových stránok na strane servera. Jeho nástupca a pokračovateľ ASP.NET je technológia pre vytváranie webových aplikácií a a služieb, ktorá je súčasťou platformy Microsoft .NET. Súčasná platforma .NET verzie 3.5 poskytuje výpelnú podporu pre webové služby s automatickou generáciou WSDL súborov. Navyše obsahuje programovací model nazvaný Windows Communication Foundation (WCF), ktorý umožňuje vývojárom vytvárať bezpečné a spoľahlivé aplikácie orientované na služby.

**Realizácia** klientskej aplikácie spočíva v použití vývojového prostredia Visual Studio vytvorením nového projektu typu ASP.NET Web Application. Do tohto projektu je potrebné pridať webovú referenciu, kde v príslušnom dialógovom okne pridáme URL adresu pre WSDL súbor obsahujúci popis webovej služby. Po pomenovaní a pridaní webovej referencie nám vývojové prostredie vygeneruje príslušné triedy pre prácu s webovou službou a jej metódami. Pre volanie webovej služby je ďalej potrebné vytvoriť inštanciu objektu, ktorá bude slúžiť ako proxy. Vytvorenie inštancie má tvar:

```
SluzbaTermProp sluzba = new SluzbaTermProp();
```

Volanie metód služby pomocou tejto inštancie prebieha rovnako ako u iných objektov pomocou bodkovej notácie s následným menom metódy a jej vstupnými parametrami:

```
label.Text = sluzba.getTermProp(regime, id, temp);
```

Keďže webová služba má návratovú hodnotu typu reťazec, výsledok volania metódy `getTermProp(regime, id, temp)` je možné zobrazit' pomocou objektu typu `Label`.

**Požiadavky.** Pre tvorbu klientskej aplikácie je požadované prostredie .NET framework 2.0, v prípade použitia WCF .NET framework verzie 3.0 a vyššie. Ako vývojové prostredie postačuje ktorákoľvek súčasná verzia Visual Studia, vrátane voľne dostupných Express verzií.

**Výhody.** Platforma .NET je voľne dostupná pre prostredie operačných systémov Windows, pričom pre Linux a ostatné operačné systémy je potrebné použiť open source platformu Mono. Vývojové prostredie Visual Studio umožňuje vývojárom rýchle vytváranie aplikácií založených na webových službách bez nutnosti poznať syntax WSDL súboru, pričom na základe popisu webovej služby z WSDL automaticky generuje triedy objektov. V porovnaní s vyššie spomenutými programovacími prostrediami umožňuje najlepšie možnosti ladenia aplikácií.

**Nevýhody.** Hlavnou nevýhodou platformy .NET je jej zameranie na prostredie Windows. Pre použitie v prostredí iných operačných systémov je potrebné použiť platformu Mono, tá však neobsahuje všetky knižnice platformy .NET. Vývojové prostredie Visual Studio je poskytované za poplatok, pričom jeho zadarmo dostupné Express verzie neobsahujú úplnú funkcionality základnej platenej verzie.

### 2.4 Klient Matlab

MATLAB je integrované prostredie pre vedecko-technické výpočty, modelovanie, návrhy algoritmov, simulácie, analýzu a prezentáciu údajov, meranie a spracovanie signálov, návrhy riadiacich a komunikačných systémov. MATLAB je nástroj tak pre pohodlnú interaktívnu prácu, ako aj pre vývoj širokého spektra aplikácií.

Realizácia klientskej aplikácie je založená v prvom kroku na vytvorení inštancie objektu obdobne ako je to v prípade PHP.

```
url = 'http://omega.tuke.sk/pavel.horovcak/php_ws/wsdl/termprop.wsdl';
createClassFromWsdL(url);
service = SluzbaTermProp;
```

Druhým krokom je volanie potrebnej funkcie webovej služby s príslušnými hodnotami vstupných parametrov funkcie

```
rez = '5'; id = '8'; t = '300';
result = getTermProp(service, rez, id, t);
```

príčom `rez` je režim práce funkcie, `id` je identifikácia zadanej látky a `t` je teplota.

Pretože webová služba vracia ako výstupnú hodnotu XML štruktúru je priamy výpis jednoduchým spôsobom jej prezentácie `disp(result)`;

Pre ďalšie použitie číselných hodnôt je potrebné štruktúru XML uloženú v premennej `result` ako pole znakov previesť na číselné hodnoty. V súvislosti s tým je potrebné do štruktúry vložiť počiatočnú a koncovú značku `container`. Následne previesť pole znakov do štruktúry v premennej `xmlR` pomocou funkcie `xml_parse` a nakoniec previesť príslušnú položku na číslo podľa nasledujúceho kódu

```
result=strcat('<container>',getTermProp(service, rez, id, t), '</container>');
```

```
xmlR = xml_parse(result);
```

```
A = {xmlR(1,1).response.data.heat_of_formation};
```

```
X = str2double(A);
```

**Požiadavky.** Pre tvorbu klienta je potrebný MATLAB od verzie 7.0.1. V prípade práce s XML štruktúrou je potrebná voľne dostupná knižnica funkcií „`xml_toolbox`“.

**Výhody.** Použitie MATLABu je veľmi vhodné hlavne vzhľadom na ďalšie budovanie vedecko-technických výpočtov, resp. matematických modelov procesov, pretože MATLAB má veľmi rozsiahlu podporu v podobe nástrojov a metód na riešenie matematických modelov, ďalej poskytuje jednoduchú možnosť zobrazenia najrôznejších grafických výstupov a v neposlednej miere umožňuje tvorbu grafického používateľského rozhrania ako napr. Delphi a iné prostriedky RAD. Výhodou je tiež možnosť použitia MATLAB Compiler pre vytvorenie samostatne spustiteľného súboru aplikácie.

**Nevýhody.** Jednou z nevýhod použitia MATLABu je, že ide o komerčný produkt. Ďalšou je skutočnosť, že musí byť na počítači nainštalovaný.

### 2.5 Java klient

Java je dobre známa otvorená platforma pre desktopové a webové aplikácie vyvíjaná spoločnosťou Oracle (predtým spoločnosťou Sun).

**Realizácia** klienta webovej služby je uskutočnená vo vývojovom prostredí Netbeans, v ktorom je potrebné vytvoriť nový projekt typu „*Web Application*“. Tento typ projektu vytvára základnú štruktúru pre knižnicu klienta. V prvom kroku je teda potrebné vytvoriť na základe WSDL opisu už zmienené knižnice. Tie je možné vygenerovať v prostredí Netbeans pomocou sprievodcu „*New Web Service Client*“, ktorého vstupom je URL adresa WSDL súboru, alebo priamo pomocou príkazu „`wsimport`“, ktorý je súčasťou JDK. Po vygenerovaní knižníc je vytvorenie klienta webovej služby relatívne jednoduché:

```
TermPropService service = new TermPropService();
```

Ďalším krokom je získanie portu webovej služby:

```
TermPropTypPortu port = service.getTermPropPort();
```

Pomocou portu je možné volať metódy webovej služby:

```
String result = port.getTermProp(regime, id, temp);
```

Keďže výstup webovej služby je definovaný ako reťazec obsahom ktorého sú XML dáta, jeho spracovanie môže byť v režii DOM:

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
```

```
DocumentBuilder builder = factory.newDocumentBuilder();
```

```
Document doc = builder.parse(new InputSource(new StringReader(result)));
```

Po vytvorení objektu typu `org.w3c.dom.Document` je prístup k hodnotám jednotlivých elementov reprezentujúcich veličiny tepelného procesu pomocou metód `getElementsByTagName` a `getTextContent`:

```
String cp = doc.getElementsByTagName("c").item(0).getTextContent();
```

**Požiadavky.** Pre realizáciu klienta je potrebná inštalácia prostredia Java 6 SDK a pre zefektívnenie programovania vývojové prostredie NetBeans alebo Eclipse.

**Výhody.** Java poskytuje otvorené a multiplatformné prostredie pre efektívnu tvorbu klienta webovej služby pomocou automatického vytvárania knižníc na základe opisu služby vo WSDL.

**Nevýhody.** Pre webové služby založené na RPC je potrebné stiahnuť novšiu verziu knižnice *wsimport* – v2.1, ktorá nie je súčasťou JDK (JDK obsahuje verziu 2.0).

## Záver

Modelovanie procesov s využitím jednej alebo viacerých webových služieb je veľmi perspektívne a širokospektrálne. Zahŕňa získavanie celého spektra údajov pre účely modelovania, začínajúc od jednej konkrétnej konštanty danej látky cez rôzne vypočítané funkčné hodnoty tejto látky až po získanie rôznych výpočtových koeficientov, resp. získanie priebehov vypočítaných funkčných hodnôt. Na druhej strane ruka v ruke s týmito požiadavkami musí byť realizovaný vývoj, formulácia a rozvoj príslušných webových služieb pre oblasť modelovania procesov. S tým súvisí rad ďalších úloh, ako rozpracovanie ošetrovania chybových stavov a metód ich prenosu na klienta, riešenie možností využívania webovej služby na strane klienta, dopĺňovanie ďalších funkcií webovej služby, rozširovanie databázovej štruktúry webovej služby.

Využívanie webovej služby v rôznych vývojových prostrediach v súčasnosti je podporované na veľmi prijateľnej úrovni. Pokiaľ ide o internetových klientov, prakticky všetky oblasti sú podporované aj open source produktami a tým sú dostupné pre každého záujemcu. V oblasti desktopových klientov je tiež výrazná podpora využívania webových služieb pri tvorbe modelov procesov. Ilustračný príklad využitia autormi navrhutej a vytvorenej webovej služby a jej volania v prostredí PHP je umiestnený na URL [http://omega.tuke.sk/pavel.horovcak/php\\_ws/phom/klienttermprop.php](http://omega.tuke.sk/pavel.horovcak/php_ws/phom/klienttermprop.php) a podobne na URL [http://omega.tuke.sk/pavel.horovcak/php\\_ws/ajaxwsom/termprop/klienttermprop.php](http://omega.tuke.sk/pavel.horovcak/php_ws/ajaxwsom/termprop/klienttermprop.php) je ukážka využitia AJAX technológie.

Príspevok bol riešený v rámci projektov VEGA 1/4194/07, 1/0194/08, 1/0365/08 a 1/0404/08.

## Literatúra

- [1] LODES, A., LANGFELDER, I. *Procesy a zariadenia I*. Bratislava: Alfa, 1987. ISBN 80-05-00150-9.
- [2] MIKLEŠ, J., FIKAR, M. *Modelovanie, identifikácia a riadenie procesov I: Modely a dynamické charakteristiky spojitého procesu*. Bratislava: STU, 1999. ISBN 80-227-1289-2.
- [3] NOSKIEVIČ, P. *Modelování a identifikace systémů*. Ostrava: Montanex, 1979. ISBN 80-7225-030-2.
- [4] VÍTEK, V. *Modelovanie a simulácia systémov I*. Bratislava: Alfa, 1981.
- [5] GEBHART, B. *Heat Conduction and Mass Diffusion*. New York: McGraw-Hill, Inc., 1993.
- [6] KAYS, W.M. a M.E., Crawford. *Convective Heat and Mass Transfer*. New York: McGraw-Hill, Inc., 1993.
- [7] DARIE, C. et al. *AJAX a PHP tvoříme interaktivní webové aplikace profesionálně*. Brno: Zoner software s.r.o, 2006. p. 320. ISBN 80-86815-47-1.
- [8] THAU, D. *Velký průvodce JavaScriptem, tvorba interaktivních webových stránek v praxi*. Praha: Grada Publishing, a.s., 2009. s. 520. ISBN 978-80-247-2211-5.
- [9] ZAKAS, N., MCPEAK, J., FAWCETT, J.. *Ajax profesionálně*. Brno: Zoner press, 2007. s. 668. ISBN 978-80-86815-77-0.
- [10] HOLZNER, S. *Mistrovství v AJAXu*. Brno: Computer Press, a.s., 2007. s. 591. ISBN 978-80-251-1850-4.

## Abstract:

SELECTED CLIENTS TECHNOLOGIES IN PROCESSES MODELING USING WEB SERVICE

**Summary:** Contribution deals with possibilities of service oriented architecture utilization in the form of web service providing selected thermodynamic data and calculations for purposes of transfer process modeling, accumulation and thermal energy transformation. There are characterized and described basic principles of transfer processes, accumulation and thermal energy transformation from the modeling point of view and requisites of thermodynamic data for mathematic models creation in the introduction of the contribution. Next part of contribution deals with procedures of web service implementation for purposes of mathematical models in typical developing (internet and desktop) environments, e.g. PHP, AJAX, ASP, MATLAB, whereby there are compared individual technologies with regard to advantages and disadvantages of their application for processes modeling.

**Key words:** processes modeling, SOA, web service, PHP, AJAX, ASP, MATLAB