

**Pavel HOROVČÁK<sup>1</sup>, Peter CIRBES<sup>2</sup>, Patrik FLEGNER<sup>3</sup>**

**VYUŽITIE SERVISNE ORIENTOANEJ ARCHITEKTÚRY PRI VÝBERE  
VLASTNOSTÍ MINERÁLOV**

**SERVICE ORIENTED ARCHITECTURE APPLICATION ON MINERALS  
PROPERTIES SELECTION**

**Abstract**

Continually amplification of internet technologies development and implementation brings possibility of productive, efficient, rich useful and interactive web applications creation. Within the latest approaches lies a SOA (Service Oriented Architecture), WS (Web Service) and AJAX (Asynchronous JavaScript And XML). The paper illustrates advantages of AJAX and WS integration on application example for interactive selection of one or more minerals according to actual chosen selection criteria. The application makes use of two web services on the server side and one web service plus Ajax technology on the client's side.

**Key words:** SOA, Web Services, Ajax, minerals, integration, PHP

**Úvod**

SOA je v súčasnosti veľmi frekventovaný pojem podporovaný asi všetkými významnými vývojovými firmami a vysvetľovaný na obrovskom množstve stránok, článkov, kníh, príspevkov a definícií. Jednou z možností realizácie služby v architektúre SOA je webová služba – WS (Samtani, 2002). WS umožňuje interoperabilitu medzi aplikáciami, ktoré pracujú na rôznych platformách. Sú špecifikované pomocou WSDL (Web Services Description Language) a výmena správ medzi klientom a serverom služby sa uskutočňuje s využitím SOAP (Simple Object Access Protocol). Ajax predstavuje kompozíciu štandardných technológií JavaScript, XML, CSS (Cascading Style Sheets), DOM (Document Object Model) a XSLT (Extensible Stylesheet Language Transformation). Integrácia týchto technológií znásobuje ich prednosti. V príspevku je tento prístup ilustrovaný na príklade výberu vlastností minerálov v aplikácii zostavenej v zmysle zásad architektúry orientovanej na služby. Aplikácia využíva prepojenie technológie Ajaxu a webových služieb.

**SOA**

Služby ako opakovane použiteľné elementy sú pokladané za softvérové stavebnicové prvky, ktoré vykonávajú špecifické funkcie. Podľa (Exforsys, 2008) neexistuje univerzálne odsúhlasená definícia SOA okrem toho, že je to architektúra, ktorá je doslova založená na orientácii na služby ako na jej hlavnom konštrukčnom. SOA v mnohých ohľadoch predstavuje evolúciu architektúry klient – server, pričom dochádza vlastne k distribúcii aplikačnej logiky na služby (na rozdiel od monolitického

---

<sup>1</sup> Doc. Ing. CSc., Technická univerzita v Košiciach, FBERG, Ústav riadenia a informatizácie výrobných procesov, B. Němcovej 3, 040 00 Košice, Slovenská republika, tel. (421 55) 602 5176, e-mail: Pavel.Horovcak@tuke.sk

<sup>2</sup> Ing., TU, FBERG, ÚRaIVP, 040 00 Košice, Slovenská republika, (421 55) 602 5174, Peter.Cirbes@tuke.sk

<sup>3</sup> Ing. PhD., TU, FBERG, ÚRaIVP, 040 00 Košice, Slovenská republika, (421 55) 602 5174, Patrik.Flegner@tuke.sk

aplikačného servera v klient-server architektúre). Systém založený na SOA môže jasne a jednoducho inkorporovať služby bežiacie na viacerých rôznych softvérových platformách. SOA berie do úvahy požiadavky poskytovateľov ako aj používateľov služby. Poskytovateľ služby ponúka jej funkcionality vo forme interfejsu k vybaveniu danej služby. Používateľ služby potom využíva toto vybavenie. Používateľom môže byť aplikácia alebo dokonca poskytovateľ inej služby. Základná vlastnosť kvalitných služieb je možnosť ich zdieľania, čo šetrí zdroje pri ich vývoji a zjednodušuje údržbu.

Popri nesporných pozitívach SOA sú známe aj oblasti a situácie, ktoré nie sú vhodné pre aplikácie SOA. Nasadenie servisne orientovanej architektúry sa neodporúča (Labanc, 2008) napríklad v týchto prípadoch: a) keď samostatné nedistribúované aplikácie nevyžadujú integráciu s inými aplikáciami či komponentami, b) keď homogénne aplikačné prostredie je postavené na jednotnej architektúre (napríklad J2EE), kde nie je potrebné zavádzať komunikáciu pomocou XML cez http, lebo existujúci protokol Java RMI je vhodnejší a preto postačujúci, c) keď ide o krátkodobou používané prototypy, demá alebo utility bez potreby rozšírenia alebo opätovného použitia v budúcnosti, d) keď aplikácie s grafickým používateľským rozhraním vyžadujú rýchlu reakciu v krátkych intervaloch.

Medzi najčastejšie opakované chyby pri zavádzaní SOA podľa spoločnosti Gartner (Pezzini, 2007) patrí napríklad a) nasadenie SOA pred dokončením výkonných celkov, čo vedie k veľkému množstvu služieb (servisov) nekorešpondujúcich s obchodným modelom a vynucujúcich si časté zmeny špecifikácie, b) opomenutie dátovej vrstvy môže mať za následok nekonzistentnosť a problémy s integritou modelu, c) prenechanie konceptu SOA len technickým pracovníkom môže viesť ku síce technologicky kvalitnému, ale obchodné požiadavky nespĺňajúcemu produktu, d) pri používaní komponentov sa stretávame s nedôverou k cudzím (mimo vývojový tím vyvinutým) celkom, dôsledkom čoho je často duplicitná práca a plytvanie zdrojmi, e) prechod na SOA bez predchádzajúceho plánovania a prípravy býva často osudný, f) SOA je rôzne chápaná rôznymi profesiami (programátori, architekti, technickí a výkonní vedúci), z čoho plynú často vážne komunikačné problémy, g) vo firme využívajúcej SOA by malo existovať koordinačné centrum pre spoluprácu jednotlivých služieb, h) centralizácia využitia SOA v celej spoločnosti je často horším riešením ako jeho rozdelenie do oddelení, pobočiek alebo do oblastí využitia, i) je nebezpečné a organizačne nevhodné navrhovať prechod na SOA skôr ako je na to firma pripravená.

Pri aplikácii SOA v praxi je preto vyžadovaný komplexný pohľad na celý systém so všetkými jeho atribútmi – informačnými tokmi, štruktúrou údajov, technickou architektúrou ale predovšetkým podnikovými procesmi.

### **Webová služba**

Webová služba je jednou z možností realizácie služby v architektúre SOA. WS poskytuje štandardný popis svojho rozhrania vo forme WSDL (Web Services Description Language), ktoré umožňuje klientovi službu využívať. Zásadne je tento popis umiestnený na inom serveri ako samotná služba a predstavuje tak akúsi spojnicu (väzbu) medzi poskytovateľom (server) a konzumentom (klient) služby. SOA sa však nerovná iba WS, lebo síce každá WS vyhovuje SOA, ale nie každá služba v rámci SOA musí byť nutne WS. V súčasnosti k základným štandardom WS (XML, SOAP, WSDL, HTTP) pribudli ďalšie rozširujúce normy (napr. WS-ReliableMessaging, WS-Addressing, WS-Notification, WS-Security, WS-Policy, WS-Choreography), ktoré dopĺňujú využitie WS a ďalej rozširujú syntax WSDL.

## AJAX

AJAX je webová vývojová technológia (presnejšie povedané skupina technológií – XHTML, CSS, DOM, XML, XSLT, XMLHttpRequest a JavaScript) určená na vytváranie interaktívnych webových aplikácií. Predstavuje ďalší logický krok vo vývoji SOA – architektúry orientovanej na služby. Pomocou Ajaxu môžu používateľské rozhrania cez prehliadač využívať webové služby ako svoj dátový zdroj na ukladanie a obnovovanie informácií (Darie et al, 2006). Dôležitá zložka Ajaxu – objekt XMLHttpRequest (XHR) – je časťou Internet Explorera 5 (od r. 1999) ako ovládací prvok ActiveX. Implementáciou XHR v ďalších prehliadačoch (Mozilla, Safari), jeho zaradením v DOM Level 3 a najmä masívnym využívaním v populárnych aplikáciách (Google Maps, Google Suggest, Gmail atď.) sa stáva XHR prakticky štandardom. Ako prvý použil termín Ajax Jesse James Garrett (z Adaptive Paths) vo februári 2005 v práci Ajax: A new Approach to Web Application (Garret, 2005). Ajax dnes zastrešuje všetky technológie podporujúce asynchrónnu komunikáciu prehliadača so serverom bez nutnosti obnovy aktuálnej stránky.

### Aplikácia výberu vlastností minerálov s využitím SOA

Minerál je pre potreby ilustrácie SOA reprezentovaný výberom štyroch jeho charakteristických vlastností –tvrdosť, trieda, kryštalografická sústava a chemický vzorec. Dátový model aplikácie je tvorený jednou tabuľkou s piatimi stĺpcami (pribúda ešte názov) a troma číselníkmi s názvami minerálov, tried a kryštalografických sústav. Funkcie výberu sú definované ako jednoparametrové (názov, tvrdosť, trieda, kryštalografická sústava), dvojpametrové (tvrdosť a trieda, tvrdosť a sústava, trieda a sústava) prípadne trojpametrové (tvrdosť, trieda a sústava). Vybraný minerál alebo skupina minerálov je prezentovaná všetkými vyššie uvedenými vlastnosťami vo zvolenej jazykovej mutácii.

### Popis webových služieb využitých v aplikácii

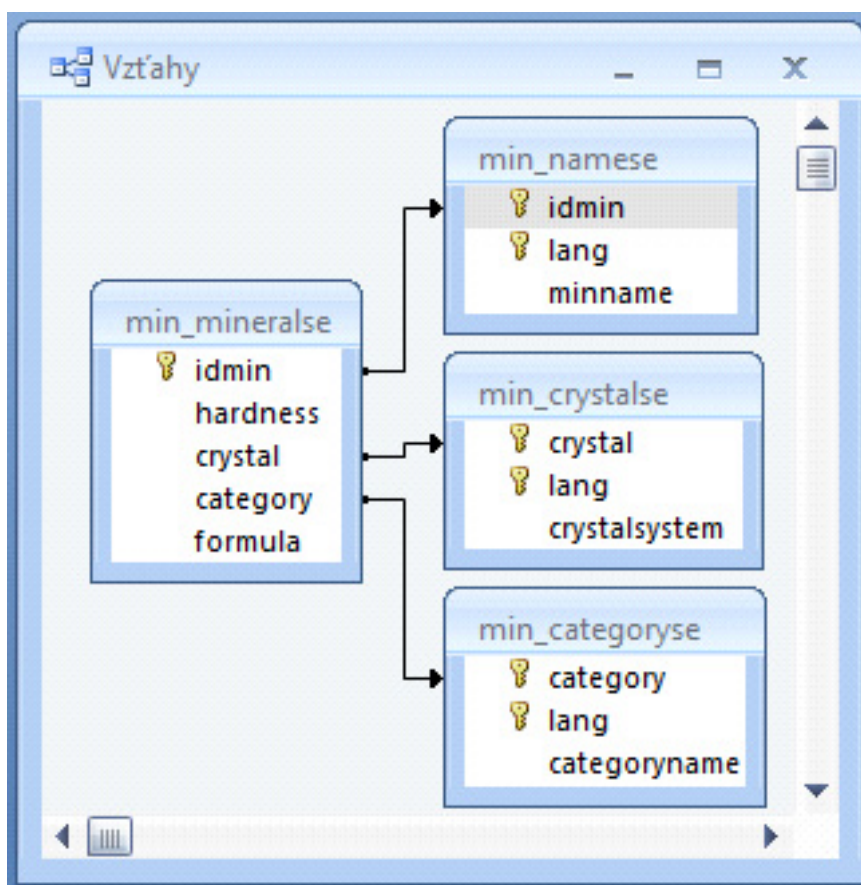
Aplikácia je zostavená s priamym alebo nepriamym využitím troch webových služieb WS1, WS2 a WS3 (podľa Obr. 3), ktoré poskytujú čiastkové funkcie.

Webová služba WS1 (SluzbaTable) je služba určená na realizáciu výberu zadaných položiek databázovej tabuľky. Účel služby je vytvorenie výberového zoznamu vo vstupnom formulári webovej aplikácie aj s možnosťou zadania parametrov pre bližšiu špecifikáciu elementu <select>. Primárne je určená pre výber z tabuliek, primárny kľúč ktorých je tvorený dvoma stĺpcami. Umožňuje ale aj výber z tabuľky s jedným primárnym kľúčom. Služba poskytuje dve funkcie v tvare:

```
array(2) {  
  [0]=> string(72)  
  "string getTable(string $Table, string $Col1, string $Col2, string $Col3)" [1]=>  
  string(120)  
  "string getSelectTable(string $Table, string $Col1, string $Col2, string $Col3,  
  string $Par1, string $Par2, string $Par3)"  
}
```

Funkcia `getTable()` má štyri parametre s významom Table – názov databázovej tabuľky, Col1 (primárny kľúč) a Col2 sú názvy stĺpcov tabuľky, hodnoty ktorých budú vrátené ako dvojice kľúč – hodnota (a použijú sa pri konštrukcii elementu `<select>` pre položky `<option>` v tvare `<option value="kľúč">hodnota</option>`). Stĺpec Col3 udáva druhý z dvojice primárnych kľúčov. V aplikácii je druhý primárny kľúč využitý na určenie jazykovej verzie obsahu tabuľky. V prípade, že tabuľka má len jeden primárny kľúč, zadáva sa hodnota Col3 = 0. Ak je potrebné z tabuľky vybrať iba jeden stĺpec, zadáva sa jeho hodnota pre Col1 aj Col2, pričom je Col3=0.

Funkcia `getSelectTable()` má sedem parametrov. Klientovi vracia kompletne zostavený element `<select>` naplnený hodnotami kľúč – hodnota zo zadanej tabuľky. Význam prvých štyroch je rovnaký ako vo funkcii `getTable()`. Význam ďalších je nasledovný: Par1 – identifikácia elementu id, Par2 – názov funkcie (v JavaScripte) volanej pri výbere (zmene) položky, Par3 – text prvej položky elementu `<select>` bez hodnoty. Výsledok je v tvare `<select id="Par1" onchange="Par2"><option value="">Par3</option><option value="key">value</option>...</select>`.



**Obr. 1** Dátový model webovej služby SluzbaMineral

Webová služba WS2 (SluzbaMineral) je služba určená na výber jedného alebo skupiny minerálov podľa zadaných kritérií zo skupiny štyroch databázových tabuliek podľa Obr. 1. Hlavná tabuľka (min\_minerale) má pre účely aplikácie iba päť stĺpcov (minerál, tvrdosť, sústava, trieda a vzorec). Dopĺňa ju skupina troch viacjazyčných číselníkov (aktuálne dva jazyky SK a EN) pre názov minerálu, sústavy a triedy. Služba má jednu funkciu getMineral() v tvare:

```
array(1) {  
  [0]=> string(99)  
  "string getMineral(string $Regime, string $Param1, string $Param2, string  
  $Param3, string $Language)"  
}
```

Funkcia getMineral() má päť parametrov s významom Regime – režim výberu, Param1, Param2 a Param3 sú parametre výberu, Language – udáva jazykovú verziu výberu. Jednotlivé režimy výberu ilustruje Obr. 2. Služba poskytuje tri jednoparametrové výbery (pre zadanú Tvrdosť, Sústavu a Triedu), tri dvojparametrové výbery (Tvrdosť + Sústava, Tvrdosť + Trieda, Trieda + Sústava) a jeden trojparametrový výber (Tvrdosť + Sústava + Trieda). Pri výbere minerálu (režim 1) na hodnote ostatných troch parametrov nezáleží (nezohľadňujú sa). Pri nulovej hodnote všetkých štyroch parametrov funkcia vracia kompletný výber všetkých položiek tabuľky. Funkcia vracia všetkých päť stĺpcov hlavnej tabuľky pre jeden minerál či viac minerálov vyhovujúcich podmienkam výberu alebo všetky riadky tabuľky (v prípade režimu 0). Ak je výsledkom podmienok výberu prázdna množina, táto skutočnosť je indikovaná v každom stĺpci návratovej štruktúry textom Prázdny výber! (Empty set!). Funkcia využíva ako klient ďalšiu webovú službu (SluzbaChemForm) na správnu interpretáciu chemického vzorca minerálu.

Webová služba WS3 (SluzbaChemForm) je služba určená na zobrazenie chemických vzorcov. Služba realizuje prevod chemického vzorca zo štandardného ASCII tvaru do HTML formátu, ktorý zobrazuje počet atómov alebo mocenstvo iónov vo forme dolných alebo horných indexov (elementami <sup> a <sub>). Algoritmické riešenie prevodu štandardného zápisu vzorca určuje typ príslušného indexu. Služba poskytuje dve funkcie v tvare:

```
array(2) {  
  [0]=> string(35)  
  "string getChemForm(string $Formula)"  
  [1]=> string(38)  
  "string getChemFormStr(string $Formula)"  
}
```

Funkcia getChemForm() má jeden parameter – chemický vzorec. Po uskutočnení prevodu funkcia vracia výsledok klientovi v štandardnom XML tvare. Pre zaradenie výsledku do HTML kódu na strane klienta je vhodné v prostredí PHP využiť štandardnú funkciu html\_entity\_decode.

Funkcia `getChemFormStr()` má rovnako jeden parameter – chemický vzorec. Od predchádzajúcej funkcie sa líši tým, že výsledok vracia klientovi nie v tvare XML, ale vo forme reťazca. V niektorých prípadoch sa tým môže zjednodušiť spracovanie na strane klienta.

### **Štruktúra aplikácie**

Aplikácia je zostavená s využitím princípov servisne orientovanej architektúry. Štruktúra aplikácie je tvorená klientskou a serverovou stranou. Obsahom serverovej strany je skupina webových služieb, volanie ktorých uskutočňuje klientska strana prostredníctvom ich interfejsových súborov – súborov WSDL. Klientska strana aplikácie (na Obr. 3 označená symbolom HTML) pozostáva z dvoch častí – statickej a dynamickej. Statická časť je tvorená štyrmi výberovými zoznamami, pomocou ktorých používateľ formuluje požadované kritériá pre výber jedného alebo skupiny minerálov. Tieto zoznamy (elementy `<select>`) sú vytvorené webovou službou WS1 so zohľadnením nastaveného jazyka. Vytvorenie výberových zoznamov je realizované na základe odoslania zvoleného jazyka (tlačidlo Jazyk/Language). Zdrojový kód ilustruje Obr. 4. Dynamická časť je realizovaná vzájomnou spoluprácou HTML (výberové kritérium), JavaScriptu a spojovacieho skriptu (Proxy). Spojovací skript (cez interfejsový súbor WSDL2) pripája webovú službu WS2. Služba WS2 dodáva na základe výberových kritérií požadované údaje o mineráloch. Celá táto funkcionálna je na strane HTML prezentovaná pomocou elementu `<div id="myDivElement">`.

### **Funkcie aplikácie**

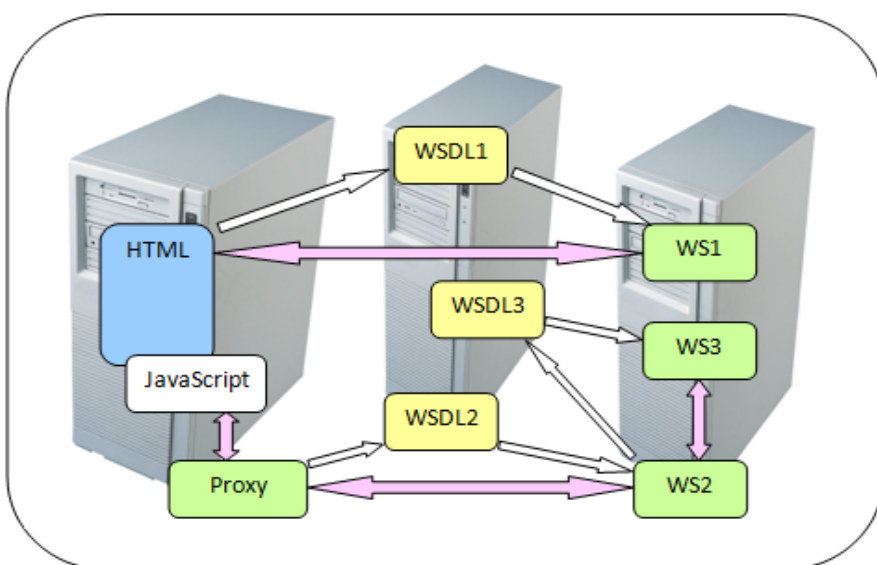
Aplikácia ponúka používateľovi jeden výberový zoznam na voľbu jazyka (teraz dva jazyky – slovenčina a angličtina). Po výbere jazyka sú k dispozícii štyri výberové zoznamy pre názov a ďalšie tri parametre minerálov (tvrdosť, kryštalografická sústava, trieda). Po výbere názvu minerálu sa zobrazia zodpovedajúce vlastnosti daného minerálu (bez ohľadu na hodnoty ostatných troch parametrov). Po výbere jedného, dvoch alebo troch parametrov minerálov sa zobrazia vlastnosti zvolenej množiny minerálov. V prípade, ak je táto množina prázdna, zobrazí sa text “Prázdny výber!”. Vzhľad obrazovky aplikácie pre konkrétny výber parametrov ilustruje Obr. 5.

Aplikácia umožňuje jednoduché rozširovanie možnosti voľby jazyka. Pridanie ďalšieho jazyka (jazykovej mutácie) do aplikácie vyžaduje niekoľko krokov. Na serverovej strane je to doplnenie troch jazykovo závislých databázových tabuliek (`min_name`, `min_crystal`, `min_category`) a doplnenie textu o prázdnom výbere vo webovej službe WS2 (premenná `$emp`). Na klientskej strane je to doplnenie názvu jazyka vo výberovom zozname v HTML (ďalšia vetva `<option>`) a odpovedajúcich textov v premenných `$head3`, `$pros` a `$h` (pozri Obr. 4) a doplnenie súboru JavaScript o hlavičku výstupnej tabuľky (premenná `var html` pre príslušný jazyk).

Konštrukcia aplikácie ilustruje súbežné využitie Ajaxu a webových služieb. Získavame tak možnosť interaktívnej práce s rôznymi zdrojmi údajov, získavaných prostredníctvom webových služieb. Tento prístup tak ilustruje možnosť praktického využívania servisne orientovanej architektúry SOA.

Služba	Režim	Par1	Par2	
Všetky hodnoty tabuľky	0	0	0	0
Minerál	1	x	y	z
Tvrdosť	2	x	0	0
Sústava	3	x	0	0
Trieda	4	x	0	0
Tvrdosť + Sústava	5	x	y	0
Tvrdosť + Trieda	6	x	y	0
Trieda + Sústava	7	x	y	0
Tvrdosť + Sústava + Trieda	8	x	y	z

**Obr. 1** Funkcie služby SluzbaMineral



**Obr. 2** Štruktúra servisne orientovanej aplikácie

[illegible]



```

    html_entity_decode($Service-
>getSelectTable("min_categoryse","category",
"categoryname",$lang,"param3","process();",$pros))."</td></tr><
/table>");
}
?>
    <div id="myDivElement">    </div>
</body>

```

**Obr. 3** Zdrojový kód klientskej časti aplikácie

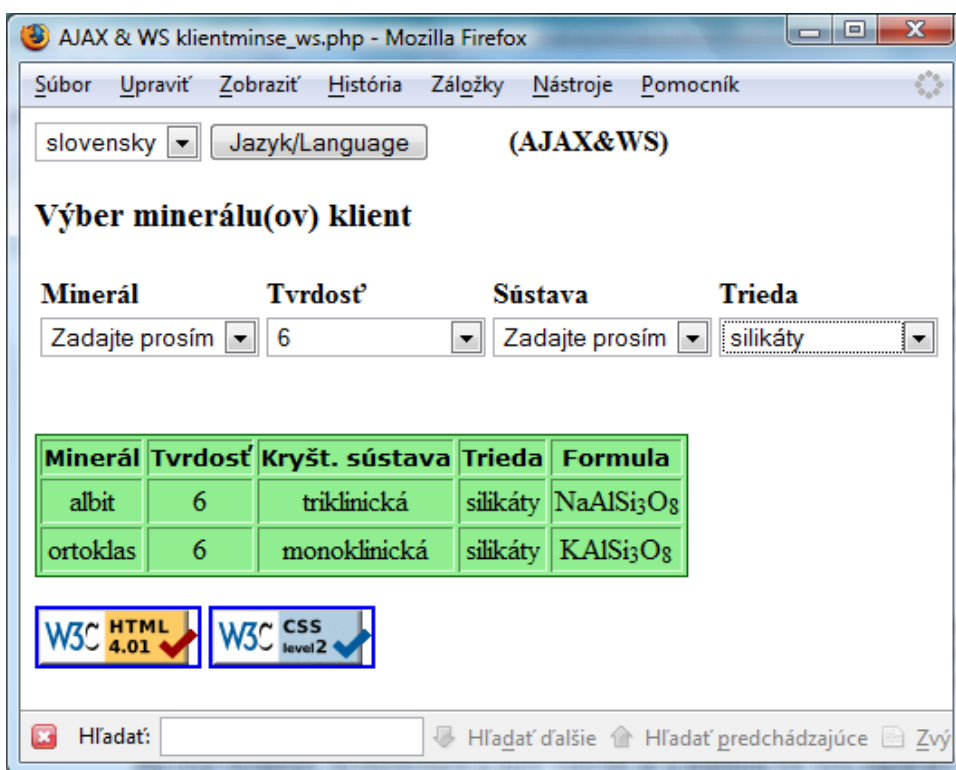
## **Záver**

Uplatňovanie prístupov a postupov SOA prináša celý rad benefitov a výhod tak pre firmu ako aj pre zostavovateľa aplikácie. Tieto výhody sa uplatňujú v jednej z nasledujúcich kategórií: štandardizované interfejsy a dátové modely, opakované používanie a spätosť.

Servisne orientovaná architektúra formou konsolidácie a opakovaným používaním aplikačných služieb umožňuje prebudovanie podnikovej infraštruktúry, odstránenie redundancií a zrýchlenie jednotlivých projektov. Umožňuje tiež jednoduchšie a rýchlejšie prispôbovanie sa zmene potrieb podniku a tiež rýchlejšiu a efektívnejšiu realizáciu nových projektov.

Spojenie asynchrónnej komunikácie medzi klientom a serverom s využívaním webových služieb prináša viaceré nové možnosti na jednej strane do výstavby aplikácie, predovšetkým rôznych vstupných obrazoviek formulárového typu, na druhej strane vedie k dekompozícii často zložitých serverových aplikácií na niekoľko jednoduchších, relatívne samostatných funkcií – služieb. Nezanedbateľnú úlohu má celý rad ďalších, na prvý pohľad neviditeľných aspektov takéhoto prístupu, medzi ktoré patrí významná redukcia množstva údajov prenášaných po sieti, z toho vyplývajúce možné zrýchlenie činnosti aplikácie (podľa niektorých výsledkov (Ahmet Fatih et al, 2007) až asi o 33%), relatívna samostatnosť “interfejsových” súborov WSDL a aj určitá zmena štandardných webových postupov návrhu a tvorby aplikácie. V mnohých prípadoch je možné dopĺňať komponenty WS, Ajaxu alebo ich kombinácie do už existujúcich aplikácií alebo projektov (inkrementálny prístup). Podstatnou výhodou je tiež možnosť prístupu k webovým službám z rôznych jazykov, vývojových prostredí aj operačných systémov.

*Príspevok bol riešený v rámci riešenia projektov VEGA 1/4194 /07 (L), VEGA 1/0194/08 (S) a VEGA 1/0365/08 (T).*



Obr. 4 Ukážka výstupnej obrazovky aplikácie

### Literatúra

- [1] Ahmed Fatih, M. and Geoffrey, F. (2007): AJAX Integration Approach for Collaborative Calendar-Server Web Services [online] 30. 4. 2007 [cited 26.3.2008] available from <[http://grids.ucsf.indiana.edu/ptliupages/publications/AJAXCCS\\_cameraready.pdf](http://grids.ucsf.indiana.edu/ptliupages/publications/AJAXCCS_cameraready.pdf)>
- [2] Darie, C., Brinzarea, B., Cherecheș-Toșa, F. a Bucica, M. (2006): AJAX a PHP tvoříme interaktivní webové aplikace profesionálně. Zoner software s.r.o Brno 2006, 320 str.
- [3] Dugáček, D. a Horovčák, P. (2008): Webová služba pre zobrazenie chemických vzorcov. In Ikaros Elektronický časopis o informační společnosti. [online]. 2008, roč. 12, č. 5 [cit. 2008-05-05]. dostupné z: <<http://www.ikaros.cz/node/4710>>. URN-NBN:cz-ik4710.
- [4] Exforsys (2008): SOA Service Oriented Architecture [online] © 2008 Exforsys.com [cited 26.3.2008] dostupné z <<http://www.exforsys.com/tutorials/soa.html>>
- [5] Garret, J.J. (2005): Ajax: A New Approach to Web Applications, [online] february 18, 2005 [cited 7.04.2008] dostupné z <<http://adaptivepath.com/ideas/essays/archives/000385.php>>

- [6] Champion, M., Ferris, Ch., Newcomer, E., Orchard, D. (2002): Web Services Architecture. W3C Working Draft 14, [online] November 2002 [cited 7.4.2008] dostupné z <<http://www.w3.org/TR/2002/WD-ws-arch-20021114/>>
- [7] Labanc, M. (2008): Tienisté stránky SOA. In: Infoware č. 3 2008, str. 14. Digital Visions s.r.o., Bratislava.
- [8] Pezzini, M. (2007): SOA's deadly dozen. In Computing, 22 Nov 2007 [online] [cited 26.3.2008] dostupné z <<http://www.computing.co.uk/computing/analysis/2203880/soa-deadly-dozen-3648889>>
- [9] Samtani, G. (2002): Top five Web service myths. Builder – Architect - WServices [online] Aug. 2002 [cited 7.4.2008] dostupné z <<http://builder.com.com/article.jhtml;jsessionid=4VPOR43OSAE21TQQACQSFFA?id=u00320020820GXS01.htm&page=2>>
- [10] Smutný, P. (2006): The Implementation Of Advanced Technologies In Intranet And Internet Web Portals. In: Proceedings of 7th International Carpathian Control Conference. Czech Republic: Rožnov p. Radhoštěm. May 29-31, 529-532.
- [11] Štumpf, J. (2007): Proč a jak zavádět architekturu SOA. In: IT Systems, CCB s.r.o Brno, ročník 9, č.10, 30 – 32; č. 9, 12 – 15; č. 9, 46 – 48.
- [12] Mustacoglu, A. F. and Fox, G. (2007): AJAX Integration Approach for Collaborative Calendar-Server Web Services [online] 30. 4. 2007 [cited 26.3.2008] available from <[http://grids.ucs.indiana.edu/ptliupages/publications/AJAXCCS\\_cameraready.pdf](http://grids.ucs.indiana.edu/ptliupages/publications/AJAXCCS_cameraready.pdf)>
- [13] Darie, Cristian – BRINZAREA, Bogdan – CHERECHEȘ-TOȘA, Filip – BUCICA, Mihai 2006. AJAX a PHP tvoříme interaktivní webové aplikace profesionálně. Zoner software s.r.o Brno 2006, ISBN 80-86815-47-1, 320 str.
- [14] Dugáček, Dušan - HOROVČÁK, Pavel 2008. Webová služba pre zobrazenie chemických vzorcov. In Ikaros Elektronický časopis o informační společnosti. [online]. 2008, roč. 12, č. 5 [cit. 2008-05-05]. dostupné z: <<http://www.ikaros.cz/node/4710>>. URN-NBN:cz-ik4710. ISSN 1212-5075.
- [15] EXFORSYS 2008. SOA Service Oriented Architecture [online] © 2008 Exforsys.com [cited 26.3.2008] dostupné z <<http://www.exforsys.com/tutorials/soa.html>>
- [16] Garret, Jesse James. 2005. Ajax: A New Approach to Web Applications, [online] february 18, 2005 [cited 7.04.2008] dostupné z <<http://adaptivepath.com/ideas/essays/archives/000385.php>>
- [17] CHampion, Michael, - FERRIS, Chris - NEWCOMER, Eric - ORCHARD, David 2002. Web Services Architecture. W3C Working Draft 14, [online] November 2002 [cited 7.4.2008] dostupné z <<http://www.w3.org/TR/2002/WD-ws-arch-20021114/>>
- [18] Labanc, Martin (2008): Tienisté stránky SOA. In Infoware č. 3 2008, str. 14. Digital Visions s.r.o., Bratislava, ISSN 1335-4787
- [19] Pezzini, Massimo (2007): SOA's deadly dozen. In Computing, 22 Nov 2007 [online] [cited 26.3.2008] dostupné z <<http://www.computing.co.uk/computing/analysis/2203880/soa-deadly-dozen-3648889>>
- [20] Samtani, Gunjan (2002): Top five Web service myths. Builder – Architect - WServices [online] Aug. 2002 [cited 7.4.2008] dostupné z

- <<http://builder.com.com/article.jhtml;jsessionid=4VPOR43OSAE21TQQACQSF>  
FA?id=u00320020820GXS01.htm&page=2>
- [21] Smutný, Pavel (2006): The Implementation Of Advanced Technologies In Intranet And Internet Web Portals. In Proceedings of 7th International Carpathian Control Conference. Czech Republic: Rožnov p. Radhoštěm. May 29-31, 2006, pp. 529-532. ISBN 80-248-1066-2
- [22] Štumpf, Jindřich (2007): Proč a jak zavádět architekturu SOA, In IT Systems, CCB s.r.o Brno, ročník 9, 2007, ISSN 1212-4567, č.10, str. 30 – 32, č. 9, str. 12 - 15, č. 9, str. 46 - 48