

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ
TECHNOLÓGIE



ÚSTAV INFORMATIZÁCIE, AUTOMATIZÁCIE A MATEMATIKY
Oddelenie informatizácie a riadenia procesov

Využitie WSDL služieb v prostredí Matlab

Bakalárska práca

Vypracovala: Andrea Szabová
Vedúci práce: Ing. Michal Kvasnica, PhD.

Bratislava 2008

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Michalovi Kvasnicovi, PhD. za pomoc, za odborné vedenie, rady a pripomienky, ktoré mi poskytoval pri vypracovávaní bakalárskej práce.

Abstrakt

Bakalárska práca rozoberá tematiku využitia webových služieb v Matlabe. Cieľom tejto práce je zoznámenie s technológiou WSDL (Web Service Description Language), analýza dostupných webových služieb a znázornenie práce s WSDL službami v Matlabe. Prvým krokom pri práci s webovými službami v Matlabe je vytvorenie objektu reprezentujúceho danú službu, za tým nasleduje samotný proces volania danej služby.

Ďalším cieľom práce je vytvoriť skript v Matlabe, v ktorom sa použije viac služieb súčasne. Prvou použitou službou je *opencalais.com*, ktorá je východiskom pre celú aplikáciu a slúži na semantickú analýzu novinových správ. Následne sa spracujú výsledky z *opencalais.com* pomocou ostatných webových služieb.

Abstract

The bachelor's degree project deals with the topic of using web services in Matlab. The aim of this project is the introduction into WSDL technology (Web Service Description Language), the analysis of available services and demonstration of work with WSDL services in Matlab. The work with services in Matlab starts with creation of WSDL-representing objects, then next step is the process of calling the web service.

Another aim of this work is the creation of a script which uses more services at once. The first service used is *opencalais.com* which is used for semantical analysis of news records. The results of this service are subsequently processed by additional web services.

Obsah

ÚVOD.....	6
1. TEORETICKÁ ČASŤ.....	7
1.1 Webové služby.....	7
1.2 XML (eXtensible Markup Language).....	8
1.2.1 Syntax XML.....	8
1.2.1.1 Základné syntaktické pravidlá.....	9
1.3 SOAP (Simple Object Access Protocol).....	9
1.4 WSDL (Web Services Description Language).....	10
2. HĽADANIE A POPIS WEBOVÝCH SLUŽIEB.....	12
2.1 Hľadanie sieťových služieb na internete.....	12
2.2 Použitie sieťových služieb v Matlabe.....	16
2.3 Calais.....	18
2.4 CountryInfoService.....	21
2.4.1 Funkcie CountryInfoService.....	22
2.5 CurrencyConvertor.....	24
2.6 CaptchaService.....	25
2.7 TextCasing.....	26
2.7.1 Funkcie TextCasing.....	27
3. POUŽITIE VIACERÝCH SLUŽIEB.....	29
3.1 Vzájomné prepojenie služieb vo forme aplikácie.....	29
3.1.1 Stránka v Matlabe.....	35
ZÁVER.....	39
LITERATÚRA.....	40

ÚVOD

V dnešnej dobe je internet jedným z najhľadanejších prameňov informácií. Okrem poskytovania informácií umožňuje komunikáciu nielen medzi ľuďmi, ale aj medzi strojmi. Komunikáciu strojov podporujú napr. webové služby, ktoré umožňujú výmenu dát medzi aplikáciami. Použitím webových služieb možno ušetriť veľa času a energie, totiž poskytujú informácie, ktorých nájdenie by zabralo veľa času a riešia rôzne úlohy, často také, na riešenie ktorých väčšina ľudí nemá k dispozícii vhodné softwarové prostriedky. Použiť sieťovú službu znamená zavolať funkciu na vzdialenom serveri, ktorá spracuje poslané údaje a vráti výsledok.

Cieľom mojej bakalárskej práce je poukázať na možnosti využitia webových služieb v Matlabe, a tým odhaliť ďalšie funkcie poskytnuté týmto programom pre vedecko-technické výpočty. V mojej práci preto uvediem niekoľko príkladov na webové služby a následne ich opíšem. Ďalej som si vytýčila za cieľ vytvoriť skript v Matlabe, ktorý vytvorí HTML stránku, do ktorej zadaním nejakej novinovej správy sa užívateľovi vygeneruje tabuľka, ktorá poskytuje informácie o menovom kurze tých krajín, ktoré sa nachádzajú v zadanej správe. Zároveň si užívateľ môže zadať určitú sumu, ktorá sa prepočíta z jednej meny na druhú.

Prvá kapitola mojej bakalárskej práce sa zaoberá základnými technológiami, na ktorých je založená štruktúra sieťových služieb. Prvým z nich je značkovací jazyk XML, poslané správy sú napísané v tomto formáte. Druhou technológiou je WSDL, formát pre popis webových služieb. Ďalšou technológiou je SOAP komunikačný protokol určený na výmenu informácií.

V druhej kapitole práce opíšem proces hľadania webových služieb na internete a uvediem pár príkladov z nájdených služieb. Po predstavení vybratých služieb nasleduje znázornenie ich volania v Matlabe a presný opis tých služieb, ktoré sú základnými kameňmi pre plánovaný skript.

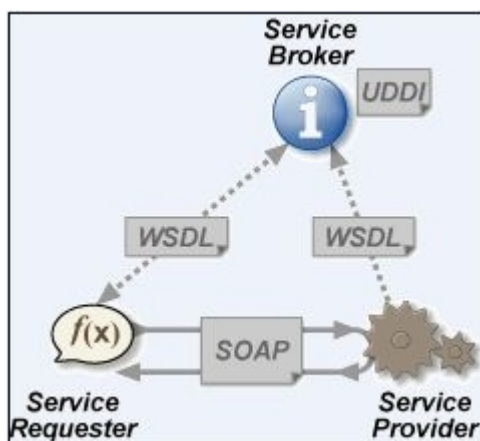
V poslednej kapitole uvediem stručný opis vytvoreného skriptu a jeho výsledok v Matlabe.

1. TEORETICKÁ ČASŤ

1.1 Webové služby^{1,2}

Webová služba je podľa definície W3C (*World Wide Web Consortium*) softwarový systém vytvorený k podpore interakcií medzi strojmi cez internet, umožňuje komunikáciu a výmenu informácií medzi aplikáciami. Štruktúra webových služieb je založená na troch základných technológiách: SOAP, UDDI, WSDL, ku ktorým sa ešte vrátim v ďalšej časti mojej práce a uvediem ich krátky opis.

Komunikácia medzi webovými službami a ostatnými systémami sa uskutoční výmenou správ XML formátu pomocou komunikačného protokolu SOAP, pričom webová služba má rozhranie popísané vo formáte WSDL.



Obr.1. Komunikácia medzi webovou službou a klientom

Pri volaní webovej služby hrá dôležitú úlohu UDDI (*Universal Description, Discovery and Integration*). UDDI je štandardný mechanizmus umožňujúci registráciu, kategorizáciu a vyhľadávanie webových služieb. Funguje ako veľký adresár obsahujúci informácie o subjektoch a nimi poskytovaných službách, pričom samotný register opäť pracuje ako webová služba.

1.2 XML (eXtensible Markup Language)^{3,4}

XML je rozšíriteľný značkovací jazyk, ktorý bol vyvinutý a štandardizovaný W3C konzorciom v roku 1998. Vytvorený bol predovšetkým pre internet, slúži na výmenu údajov medzi aplikáciami a na publikovanie dokumentov.

Samotný pojem XML sa dnes používa v troch trošku odlišných významoch:

XML je

- formát, ktorý slúži na výmenu a ukladanie dát;
- metajazyk definujúci syntax definície iných jazykov;
- celá sada technológií, ktoré úzko súvisia s jazykom XML (XSL, XLink, XPointer).

XML je na prvý pohľad podobný jazyku HTML, hoci sa od seba zásadne líšia a používajú sa k iným účelom. Podobnosť medzi nimi pochádza od rovnakých rodičov – obidva jazyky sú odvodené od jazyka SGML, napriek tomu jedným z najdôležitejších rozdielov je to, akým spôsobom sú odvodené od SGML. XML je podmnožina SGML – ľahšia verzia, ktorá bola oproti SGML zjednodušená, aby umožňovala použitie na internete, ale aj ako formát výmeny dát. Oproti tomu jazyk HTML je aplikáciou SGML – je to teda konkrétny jazyk, ktorý dodržiava štandard SGML. Ďalej XML oproti HTML nepovie nič o tom, ako má daný dokument vyzerieť, totiž XML dokumenty len popisujú štruktúru dát, ale nijak nedefinujú, ako sa majú uložené informácie prezentovať. Výhodou jazyku XML je, že autori si môžu definovať vlastné tagy, ktoré dokážu presne označiť význam prezentovaných informácií, kým HTML má preddefinované značky.

1.2.1 Syntax XML

XML dokument sa skladá z elementov. Element môže obsahovať ďalšie elementy alebo dáta. Element môže mať dané atribúty, ktoré určujú vlastnosti elementu.

Syntax elementu:

`<meno_elementu atribút="hodnota atribútu">`

dáta, ktoré sú obsahom elementu

`</meno_elementu>`

1.2.1.1 Základné syntaktické pravidlá

- Každý element musí mať uzatvárací tag:
`<element>obsah elementu</element>`,
alebo je zapísaný skrátene (v tom prípade nemôže obsahovať žiadne ďalšie elementy alebo text):
`<element/>`
- Názvy elementov rozlišujú veľké a malé písmená:
`<Zly_priklad>obsah</zly_priklad>`
`<Dobry_priklad>obsah</Dobry_priklad>`.
- Elementy sa nesmú prekrývať:
`<zly_priklad></zly_priklad>`
`<dobry_priklad></dobry_priklad>`.
- XML dokument musí mať koreňový element, všetky ostatné elementy sú v ňom obsiahnuté.
- Hodnota atribútu musí byť v úvodzovkách.
- Úvodzovky ohraničujúce hodnotu atribútu môžu byť jednoduché aj zložené:
`<element atribút="hodnota atribútu">`
`<element atribút='hodnota atribútu'>`.
- Syntax komentáru:
`<!-- komentár -->`.

Jazyk XML by mal byť použiteľný aj v budúcnosti, keď bude musieť vyhovieť ďalším požiadavkám, preto je navrhnutý tak, aby bol ľahko rozširiteľný.

1.3 SOAP (Simple Object Access Protocol)^{2,5}

SOAP je jednoduchý komunikačný protokol, určený na výmenu štruktúrovaných informácií v decentralizovanom, distribuovanom prostredí. SOAP používa technológie XML na definovanie rozširiteľného komunikačného rámca poskytujúceho štruktúru správ, ktoré môžu byť vymieňané prostredníctvom množstva základných protokolov. Rámec bol navrhnutý tak, aby bol nezávislý od každého konkrétneho programovacieho modelu a iných špecifických sémantik jednotlivých implementácií. Najčastejšie sa

SOAP používa ako náhrada RPC vzdialeného volania procedúr, keď jeden z účastníkov komunikácie je klient a druhý server, klient pošle požiadavku v XML formáte, server hneď odpovedá. Webový server čaká na požiadavky klientov, pričom v momente kedy cez HTTP príde SOAP-ová správa, spustí príslušnú webovú službu, predá jej požiadavku a výsledok zašle ako správu späť klientovi. Pomocou formátu SOAP možno realizovať aj zložitejšie komunikácie.

1.4 WSDL (Web Services Description Language)^{2,4,6}

WSDL - *Web Services Description Language* je štandardný formát pre popis sieťových služieb. Je jazykom, ktorý opisuje, aké možnosti ponúka webová služba, aké parametre majú jednotlivé metódy a aké hodnoty vrátia. Vyvinutý bol firmami Microsoft a IBM, ktoré si uvedomili potrebu zjednotenia jazyka používaného pre popis rozhraní webových služieb. WSDL sa zapisuje vo formáte XML a používa SOAP komunikáciu. Podporované operácie a správy sú opísané abstraktne a potom sa obmedzujú na konkrétny sieťový protokol a formát správy, to umožňuje ľahké vytvorenie popisu rozhraní. WSDL dokument používa nasledujúce elementy pre popis sieťových služieb:

<types> Definuje dátové typy používané webovou službou.

<message> Definuje správy používané webovou službou.

<operation> Popis akcie podporovanej webovou službou.

<portType> Definuje webovú službu, operácie a správy, ktoré sú použité webovou službou.

<binding> Definuje formát správ a podrobnosti protokolov používaných každým portom.

<port> Jednoduchý koncový bod definovaný pomocou sieťovej adresy a väzby.

<service> Množina súvisiacich koncových bodov.

WSDL dokument obsahuje popis jednej služby. Jedna služba má jednu alebo viac brán (*port*). Každá brána má väzbu (*binding*), čo je spôsob ako sa daná brána volá (napríklad SOAP-cez-HTTP), a nejakú prístupovú adresu (*URL*). Jedna služba tak teoreticky môže mať viac brán s rôznymi väzbami, tj. možno volať jednu službu rôznymi spôsobmi. Väzby odkazujú na rozhranie (*portType*), ktoré je súhrnom operácií

(*operation*). Rozhranie v objektovo-orientovaných jazykoch zodpovedá objektu, jednotlivé operácie zodpovedajú metódam objektu, alebo v jazyku C funkciám. Každá operácia definuje obvykle dve správy (*message*), jednu vstupnú a jednu výstupnú, ale môže i menej. Každá správa obsahuje žiadnu, jednu alebo viac častí (*part*), ktoré zodpovedajú parametrom a návratovým hodnotám. Z toho vyplýva, že volaná funkcia môže mať viac návratových hodnôt než len jednu. Typy parametrov a návratových hodnôt sú definované pomocou *XML Schema*.

Príklad:

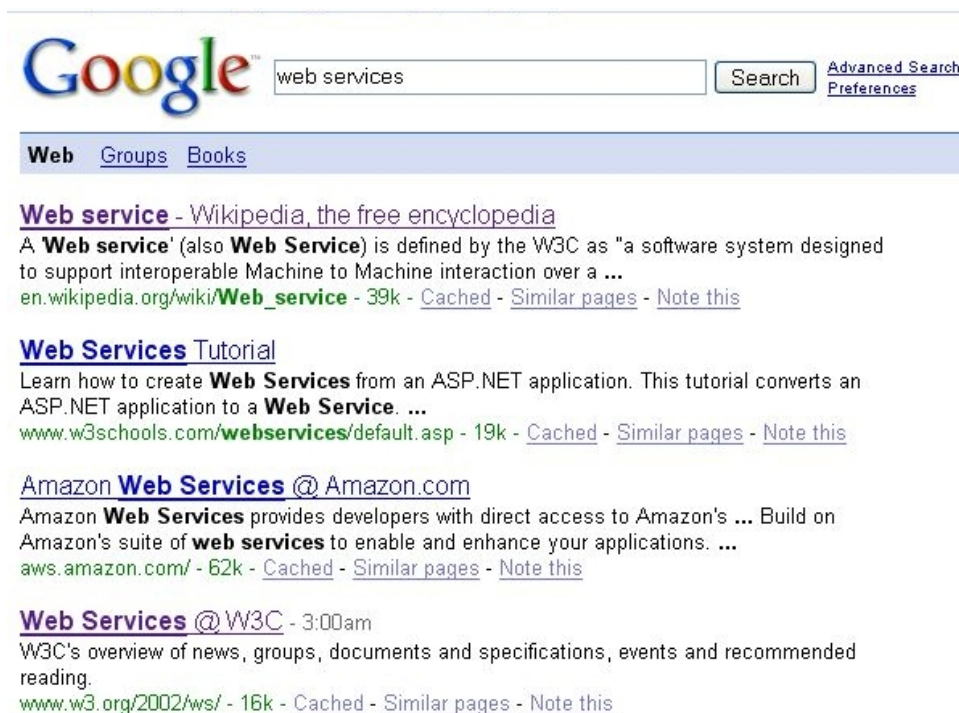
Časť WSDL dokumentu definujúca funkciu *jePrvocislo*:

```
<message name="jePrvocisloRequest">
  <part name="cislo" type="xsd:long"/>
</message>
<portType name="Cisla">
  <operation name="jePrvocislo" parametrOrder="cislo">
    <input message="m:jePrvocisloRequest" name="jePrvocisloRequest"/>
    <output message="m:jePrvocisloResponse" name="jePrvocisloResponse"/>
  </operation>
</portType>
<binding name="cislaSoapBinding" type="m:Cisla"> ... </binding>
<service name="CislaService">
  <port binding="m:cislaSoapBinding" name="cisla">
    <wsdlsoap:address location="http://nekde.cz/cisla"/>
  </port>
</service>
```

2. HLADANIE A POPIS WEBOVÝCH SLUŽIEB

2.1 Hľadanie sieťových služieb na internete

Ako som už spomenula, cieľom tejto práce je poukázať na možnosti využitia webových služieb v Matlabe pomocou skriptu obsahujúceho viac na seba nadväzujúcich služieb. Prvým krokom bolo preto nájsť vhodné služby na sieti, ktoré by mali byť základnými kameňmi skriptu. Pri hľadaní som použila internetový vyhľadávač Google. Zadaním výrazu *web services* som dostala za výsledok nasledovné adresy:



Hľadáním služieb som pokračovala na stránke www.xmethods.net, ktorá poskytuje dlhý rad webových služieb. Domovská stránka hore uvedenej URL vyzerá nasledovne:

Full Service List Also accessible via XML Interfaces: [DISCO](#) [WS-Inspection](#) [RSS](#) See the [interfaces](#) section for more information.

List is ordered by submission time, with most recent services listed first.

Publisher	Style	Service Name	Description	Implementation
VOORSPRONG	DOC	Try It: ISENTest	This webservice contains a function to validate an 10 digit and a 13 digit ISBN number	Visual Dataflex
VOORSPRONG	DOC	Try It: ElevenTest	Checksum ElifProof test, usable for Dutch Bank Account and Burger Service Numbers	Visual Dataflex
aandreu	RPC	Try It: Salted SHA Hash Generator	A SOAP service that generates salted SHA hashes.	NuSOAP
amala_raj	DOC	Try It: BibleWebservice	Get God's Word by Books,Chapter,Verse and keyword from the BIBLE	
amala_raj	DOC	Try It: Real Time Market Data	This Web Service offers various sets of data designed for all market participants - from the individual investor to market professionals. These services provide the opportunity to search INET data and have a unique perspective on the market.	
StrikeIron	RPC	Try OptionsOnTrack Historic Stock Option Data	Get historical end of day data for U.S. stock options	
CDYNE	DOC	Try It: CDYNE 411 - Directory Assistance	Whitepages, Yellow Pages, Reverse Address, Reverse Phone, Neighbors, and more!	
StrikeIron	DOC	Try The Sports Network Major League Baseball	Major League Baseball teams, schedules, statistics, injuries, scores, polls, standings and more	

Nachádza sa tu veľké množstvo rôznych služieb, vedľa ich názvov je uvedená stručná charakteristika v anglickom jazyku. Stlačením odkazu určitej služby dostaneme viac informácií o nej. Na ukážku som vybrala službu *Temperature Conversions*, ktorá podľa popisu premení °C(stupne Celsia) na °F (Fahrenheit).

VOORSPRONG	DOC	Try It: TemperatureConversions	Conversions from Celsius to Fahrenheit and vice versa	Visual Dataflex
------------	-----	--	---	-----------------

Po stlačení odkazu som sa dostala na nasledovnú stránku, kde je uvedená adresa WSDL dokumentu služby. Ďalej sa tu nachádzajú informácie, ako je vlastník, podrobnejší popis a URL samotnej služby.

Temperature Conversions

Try It

WSDL	http://webservices.daehosting.com/services/TemperatureConversions.wso?WSDL Analyze WSDL View RPC Profile (only for RPC services)
Owner:	VOORSPRONG
For more Info:	http://www.dataaccess.com
Description:	Conversions from Celsius to Fahrenheit and vice versa

Endpoints

URL	Publisher	Contact Email	Implementation
http://webservices.daelab.net/temperatureconversions/TemperatureConversions.wso	VOORSPRONG	vincent.oorsprong@dataaccess.nl	Visual Dataflex

Contributed Clients [What is this?](#) [Add / Edit / Delete Client](#)

No clients are currently listed

Detailed Description

This webservice contains temperature functions such as:

- Conversion of a celsius temperature to a fahrenheit temperature
- Conversion of a fahrenheit temperature to a celsius temperature
- Windchill temperature calculation

The functions have a minimum value detection is build in.

Zadaním URL samotnej služby do prehliadača

(<http://webservices.daelab.net/temperatureconversions/TemperatureConversions.wso>), ktorá sa nachádza pod nadpisom URL, sa objavila stránka, kde okrem vybratej služby *TemperatureConversions* boli uvedené tri ďalšie služby:

Visual DataFlex Web Services

This domain contains the following webservises

- [ElevenTest](#)
- [ISBN Tests](#)
- [Date Relates Services](#)
- [Temperature Conversion Functions](#)

Ďalej odkazom *TemperatureConversion Functions* som dostala zoznam metód služby:

Visual DataFlex Web Service for Temperature Conversions

Visual DataFlex Web Service to convert temperature values between Celcius and Fahrenheit

The following operations are available. For a formal definition, please review the [Service Description](#).

- [CelciusToFahrenheit](#)
Converts a Celcius Temperature to a Fahrenheit value
- [FahrenheitToCelcius](#)
Converts a Fahrenheit Temperature to a Celcius value
- [WindChillInCelcius](#)
Windchill temperature calculated with the formula of Steadman
- [WindChillInFahrenheit](#)
Windchill temperature calculated with the formula of Steadman

Pod odkazom *Service Description* sa nachádza WSDL dokument služby. Ostatné štyri odkazy predstavujú jednotlivé funkcie služby:

CelsiusToFahrenheit – Slúži na premenu z °C na °F.

FahrenheitToCelsius – Slúži na premenu z °F na °C.

WindChillInCelsius – Určuje zdanlivú teplotu na povrchu ľudského tela, tzv. ‘WindChill’ v °C v závislosti od rýchlosti vetra a teploty zadanej v °C.

WindChillInFahrenheit – Určuje zdanlivú teplotu na povrchu ľudského tela v °F v závislosti od rýchlosti vetra a teploty zadanej v °F. Ďalej uvediem ukážku na použitie metód *WindChillInCelsius* a *FahrenheitToCelsius*. Po stlačení odkazu na danú metódu sa objaví tabuľka, kde je potrebné zadať vstupné údaje, v prípade metódy *WindChillInCelsius* teplotu v stupňoch Celsia a rýchlosť vetra v m/s.

WindChillInCelsius

Windchill temperature calculated with the formula of Steadman

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
nCelsius	<input type="text" value="25"/>
nWindSpeed	<input type="text" value="4"/>

Po poslaní údajov dostaneme výsledok sprevádzaný XML tagmi.

`<decimal>23.3104</decimal>`

V tomto prípade odpoveď znie takto: Zdanlivá teplota sa pohybuje okolo 23 °C pri teplote 25°C a rýchlosti vetra 4 m/s.

Ďalší príklad na premenu z °F na °C:

FahrenheitToCelsius

Converts a Fahrenheit Temperature to a Celsius value

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
nFahrenheit	<input type="text" value="40"/>

Odpoveď:

`<decimal>4.444444</decimal>`

Čiže 40 °F sa rovná 4.444 °C.

2.2 Použitie sieťových služieb v Matlabe

Pri volaní služieb Matlabom sa najprv používa funkcia *createClassFromWsdL*. Táto funkcia vytvorí matlabovskú triedu, ktorá je založená na metódach webovej služby. Tu vidíme schému na vytvorenie takejto triedy, k čomu je potrebné vedieť URL WSDL dokumentu služby:

```
createClassFromWsdL('http://priklad.com/sluzba.wsdl')
```

Namiesto URL môžeme použiť aj relatívnu cestu k službe, keď ju máme k dispozícii na serveri:

```
createClassFromWsdL('\mojadresarsluzby\sluzba.wsdl')
```

Na zistenie metód poskytnutých webovou službou používame príkaz *methods*:

```
methods menosluzby
```

Samotný proces volania sa realizuje podľa nasledujúceho modelu:

```
vystup_zo_sluzby = metoda_sluzby(meno_sluzby, ziadane_udaje)
```

Pod žiadanými údajmi sa rozumie množina vstupných údajov potrebných pre generovanie výsledku a pod menom služby sa rozumie názov vytvorenej triedy v Matlabe, tzv. odpoveď vrátená po zadefinovaní WSDL dokumentu.

Ďalej uvediem konkrétny príklad na proces volania sieťovej služby v Matlabe. Na tento účel som vybrala službu *GlobalWeather*, ktorú som našla na stránke www.xmethods.com. Táto webová služba poskytuje informácie o aktuálnom počasi v určitom meste v určitej krajine. Postupovala som nasledovne:

Zadefinovala som WSDL dokument služby pomocou príkazu *createClassFromWSDL*:

```
>>createClassFromWSDL('http://www.webservice.com/globalweather.asmx?wsdl')
```

Odpoveď po vytvorení triedy pre danú službu je názov danej služby:

```
ans =  
GlobalWeather
```

Ďalej som použila príkaz *methods* na zistenie metód poskytnutých danou službou:

```
>> methods GlobalWeather  
Methods for class GlobalWeather:
```



```
GetCitiesByCountry GlobalWeather
GetWeather          display
```

Výsledkom sú dve metódy: *GetCitiesByCountry* a *GetWeather*. Vybrala som *GetWeather* a zadala som príkazový riadok podľa hore uvedenej schémy. V tomto prípade ako žiadané údaje k spracovaniu, bolo treba zadať meno mesta a krajiny. Je dôležité pripomenúť, že pri volaní webovej služby, čiže pri použití spomenutého príkazového riadku, sa reťazce v zátvorkách (okrem metód danej služby) píšú v úvodzovkách.

```
>> pocasie = GetWeather(GlobalWeather, 'Bratislava', 'Slovakia')
```

Výsledok som dostala v takejto forme:

```
pocasie =
<?xml version="1.0" encoding="utf-16"?>
<CurrentWeather>
  <Location>Bratislava Ivanka, Slovakia (LZIB) 48-12N 017-12E
142M</Location>
  <Time>May 25, 2008 - 01:30 PM EDT / 2008.05.25 1730 UTC</Time>
  <Wind> from the NNE (030 degrees) at 6 MPH (5 KT):0</Wind>
  <Visibility> greater than 7 mile(s):0</Visibility>
  <Temperature> 68 F (20 C)</Temperature>
  <DewPoint> 50 F (10 C)</DewPoint>
  <RelativeHumidity> 52%</RelativeHumidity>
  <Pressure> 30.00 in. Hg (1016 hPa)</Pressure>
  <Status>Success</Status>
</CurrentWeather>
```

V odpovedi XML formátu sú hodnoty veličín charakterizujúcich počasie uvedené medzi párovými tagmi, z ktorých je základným elementom `<CurrentWeather>`, ostatné elementy definujú:

```
<Location> - miesto merania
<Time> - čas merania
<Wind> - smer a rýchlosť vetra
<Visibility> - viditeľnosť
<Temperature> - teplotu
<DewPoint> - saturačnú teplotu
<RelativeHumidity> - vlhkosť vzduchu
<Pressure> - tlak
```

Po preskúmaní pár služieb stránky www.xmethods.net, z ktorých som sa rozhodla niektoré použiť v plánovanom skripte (napr.: *FreeCaptchaService*), som pokračovala hľadaním ďalších služieb na internete. V ďalšej časti uvediem a opíšem webové služby, ktoré hrajú dôležitú úlohu v skripte.

2.3 Calais

Webová služba *Calais* dokáže spracovaním prirodzeného jazyka, pomocou strojového učenia a iných metód analyzovať anglický text, nájsť v texte a roztriediť entity (osoby, miesta, organizácie, atď.), fakty (napr.: osoba 'x' pracuje v spoločnosti 'y') a rôzne udalosti. *Calais* podporuje komunikácie založené na SOAP a POST protokoloch.

Pred použitím služby *Calais* je potrebné sa zaregistrovať na stránke www.opencalais.com. Užívateľ tým získa kľúč, použitím ktorého môže zavolať službu denne 40 000 krát.

Pri volaní služby *Calais*, ako pre všetky webové služby zavolané v Matlabe je prvým krokom vytvoriť matlabovskú triedu, k čomu je potrebné poznať URL WSDL dokumentu služby. WSDL dokument služby *Calais* sa nachádza pod adresou: <http://api.opencalais.com/enlighten/?wsdl>. Nasledujúcim krokom je posielanie údajov na spracovanie podľa nasledujúceho vzoru zadaného na domovskej stránke *Calaisu*:

```
string Enlighten(string licenseID, string content, string paramsXML)
```

Kde *licenseID* je kľúč získaný pri registrácii, *string content* je text pre analýzu a *paramsXML* sú parametre XML, pre ktorých je uvedený príklad na spôsob ich zadefinovania pod adresou www.opencalais.com/calaisAPI.

Tabuľka 1 Vysvetlivka k modelu pre volanie služby *Calais*

Meno poľa	Typ	Definícia	Poznámka
licenseID	reťazec	Prihlasovací kľúč	Získaný registráciou
content	reťazec	Obsah na spracovanie (správa)	Max. dĺžka vstupu je 100 000 znakov
paramsXML	reťazec	Vstupujúce parametre	Max. dĺžka parametrov je 16 000 znakov

¹Tabuľka prevzatá z internetovej stránky: www.opencalais.com/calaisAPI

Text poslaný do *Calaisu* na analýzu je sprevádzaný parametrami XML, ktoré sú opísané v tabuľke 2.

Tabuľka 2

Povinné vstupné parametre

Parameter	Sekcia	Definícia	Hodnota
contentType	Processing Directives	Formát vstupujúceho textu	“TEXT/XML“, “TEXT/TXT” alebo “TEXT/HTML“
outputFormat	Processing Directives	Formát vráteného výsledku	“XML/RDF“, “Text/Simple“ alebo “Text/Microformats“
reltagBaseURL	Processing Directives	Základné URL vkladané do rel-tag mikroformátov	<základný URL>, napr.: “http://myblog.com/tag“
allowDistribution	User Directives	Ukáže či extrahované metadáta môžu byť distribuované	“true“ or “false“ (áno/nie)
allowSearch	User Directives	Ukáže či môžu byť budúce hľadania uskutočnené na extrahovaných metadátach	“true“ or “false“ (áno/nie)
externalID	User Directives	Vygenerované ID pre podanie	reťazec
submitter	User Directives	Identifikátor pre text	reťazec

²Tabuľka prevzatá z internetovej stránky: www.opencalais.com/calaisAPI

Na ukážku som vybrala správu uvedenú na domovskej stránke *Calaisu*. Na danú správu odkazuje link :

<http://www.reuters.com/article/technology-media-telco-SP/idUSNAAD300120070430?sp=true>.

```
>> createClassFromWsdL('http://api.opencalais.com/enlighten/?wsdl')
ans =
calais
>> c = calais;
>> kluc = '6dber2gs3rp75xt52pspetrs';
>> params = '<c:params xmlns:c="http://s.opencalais.com/1/pred/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<c:processingDirectives c:contentType="text/txt"
c:outputFormat="xml/rdf">
</c:processingDirectives>
<c:userDirectives c:allowDistribution="true" c:allowSearch="true"
c:externalID="17cabs901" c:submitter="ABC">
</c:userDirectives>
```

```
<c:externalMetadata>
</c:externalMetadata>
</c:params>';
```

Pri zadaní textu je dôležité zdvojnásobniť vyskytujúce sa úvodzovky, zložité úvodzovky netreba, len jednoduché:

```
>> sprava= 'Reuters to acquire text search firm ClearForest  Mon Apr
30, 2007 7:00am EDT  http://www.reuters.com/article/technology-media-
telco-SP/idUSNAAD3001200... ';
>> out = Enlighten(c, key, sprava, params);
```

Výsledok vyzerá nasledovne:

IndustryTerm: text search, broader search development, text analytics software, development group, information supplier, auto-coding tools, search techniques
Company: Autonomy Corp., Time Warner Inc., Elsevier Science, Reuters, Nstein, Xerox Corp., Pitango Venture Capital, Inxight, ClearForest Ltd., DB Capital Partners, JP Morgan Partners, Booz Allen Hamilton, Greylock Partners, ClearForest Mon Apr, Dow Chemical, HarbourVest Partners, Reuters Group Plc, Walden Israel, International Finance Corp., AOL LLC
Country: United States, Israel
URL: http://www.reuters.com/article/technology-media-telco-SP/idUSNAAD3001200
Person: Gerry Campbell, Eric Auchard
ProvinceOrState: Massachusetts
City: Tel Aviv, Waltham
Technology: business intelligence

Odpoveď má jednoduchý formát, v ktorom jednotlivé typy semantických metadát majú nasledovné funkcie:

IndustryTerm – Vyzdvihne z textu popis nejakého priemyslu, oddelenia, produktu, obchodu.

Company – Uvedie mená organizácií.

Country – Uvedie mená krajín.

URL – Uvedie URL nachadzajúci sa v zadanej správe.

Person – Nájde mená osôb v správe.

ProvinceOrState – Nájde meno provincií, štátov alebo nejakých iných politicky definovaných oblastí.

City – Uvedie mená miest.

Technology – Vyzdvihne z textu meno alebo popis nejakej technológie.

Odpoveď služby *Calais* môže mať tieto formy:

1. RDF metadáta – je formát podporujúci popis zdrojov pomocou abstraktného modelu metadát, implementovaná v jazyku XML.

Napr.: Organizácia s menom “ClearForest Ltd.” by vyzerala v RDF formáte takto:

```
<rdf:Description rdf:about="http://d.opencalais.com/comphash-1/9dd2192a-4cd2-3b9a-ac2f-b6a0d1fed773">
  <rdf:type
rdf:resource="http://s.opencalais.com/1/type/em/e/Company"/>
  <c:name>ClearForest Ltd.</c:name>
</rdf:Description>
```

2. Mikroformáty – *Calais* podporuje tri mikroformáty: *rel-tag*, *hCard* a *hCalendar*.
3. Jednoduchý formát – odpoveď má formu zoznamu, kde sú uvedené najprv metadátové prvky a potom príslušné hodnoty z výstupu, napr.:
City: Bratislava

V predchádzajúcej časti som uviedla len pár príkladov na metadáta použité v *Calaise*. Presný popis všetkých semantických metadát sa nachádza pod adresou: www.opencalais.com/calaisAPI#extractedsemanticdata.

2.4 CountryInfoService

Ďalšou použitou službou je *CountryInfoService*, ktorá poskytuje informácie (hlavné mesto, použitá mena, jazyk, atď.) o jednotlivých krajinách.

WSDL dokument služby *CountryInfoService* sa nachádza pod adresou: <http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL>.

Pomocou príkazu *methods* možno overiť veľké množstvo funkcií poskytnutých *CountryInfoServiceom*.

2.4.1 Funkcie CountryInfoService

Zoznam funkcií služby *CountryInfoService* po zadaní príkazu *methods* do Matlabu :

Methods for class CountryInfoService:

CapitalCity	LanguageName
CountriesUsingCurrency	ListOfContinentsByCode
CountryCurrency	ListOfContinentsByName
CountryFlag	ListOfCountryNamesByCode
CountryISOCode	ListOfCountryNamesByName
CountryInfoService	
ListOfCountryNamesGroupedByContinent	
CountryIntPhoneCode	ListOfCurrenciesByCode
CountryName	ListOfCurrenciesByName
CurrencyName	ListOfLanguagesByCode
FullCountryInfo	ListOfLanguagesByName
FullCountryInfoAllCountries	display
LanguageISOCode	

Opis jednotlivých funkcií:

CapitalCity – vráti hlavné mesto zadanej krajiny. Príklad v Matlabe:

```
>>createClassFromWsd('http://webservices.oorsprong.org/websamples.coun  
tryinfo/CountryInfoService.wso?WSDL');  
>> c = CountryInfoService;  
>> hlavne_mesto = CapitalCity(c, 'SK')  
hlavne_mesto =  
    Bratislava
```

CountriesUsingCurrency – vráti ISO kód a názvy krajín, v ktorých je použitá určitá mena. Pr.:

```
>> krajiny = CountriesUsingCurrency (c, 'EUR')  
ans =  
    tCountryCodeAndName: [24x1 struct]  
>> ans.tCountryCodeAndName(:)  
ans =  
24x1 struct array with fields:  
    sISOCode  
    sName
```

Odpoveď v tomto prípade je pole so dvadsaťštyrmi prvkami. Hodnoty sName sú:

Andorra, Austria, Aaland Islands, Belgium, Germany, Spain, Finland, France, French Guiana, Guadeloupe, Greece, Ireland, Italy, Luxembourg, Monaco, Martinique, Netherlands, Saint Pierre And Micquelon, Portugal, Reunion, San Marino, French Southern Territories, Vatican City, Mayotte.

Hodnoty sISOCode sú: AD, AT, AX, BE, DE, ES, FI, FR, GF, GP, GR, IE, IT, LU, MC, MQ, NL, PM, PT, RE, SM, TF, VA, YT.

CountryCurrency – vráti ISO kód a meno používanej meny krajiny. Pr.:

```
mena = CountryCurrency(c, 'SK')
ans =
    sISOCode: 'SKK'
    sName: 'Koruny'
```

CountryFlag – vráti URL na sieť pre obraz vlajky zadanej krajiny. Pr.:

```
>> vlajka = CountryFlag(c, 'SK')
vlajka =
    http://www.oorsprong.org/WebSamples/CountryInfo/Images/Slovak
    ia.jpg
```

CountryISOCode – vráti ISO kód krajiny podľa zadaného mena krajiny. Pr.:

```
>> CountryISOCode(c, 'Slovakia')
ans =
    SK
```

CountryIntPhoneCode – vráti medzinárodné smerovacie číslo danej krajiny. Pr.:

```
>> MSC = CountryIntPhoneCode(c, 'SK')
MSC =
    421
```

CountryName – vráti meno krajiny podľa zadaného ISO kódu. Pr.:

```
>> CountryName(c, 'SK')
ans =
    Slovakia
```

CurrencyName – vráti meno meny podľa zadaného ISO kódu meny. Pr.:

```
>> CurrencyName(c, 'SKK')
ans =
    Koruny
```

FullCountryInfo – vráti všetky informácie poskytnuté službou o zadanej krajine. Pr.:

```
>> vsetky_informacie = FullCountryInfo(c, 'SK')
vsetky_informacie =
    sISOCode: 'SK'
    sName: 'Slovakia'
    sCapitalCity: 'Bratislava'
    sPhoneCode: '421'
```

```

        sContinentCode: 'EU'
    sCurrencyISOCODE: 'SKK'
    sCountryFlag: [1x67 char]
    Languages: [1x1 struct]

```

LanguageName – vráti jazyk používaný v danej krajine.Pr.:

```

>> jazyk = LanguageName (c, 'SK')
jazyk =
Slovak

```

ListOfContinentsByCode – vráti zoznam mien a ISO kódov všetkých kontinentov.Pr.:

```

>> ListOfContinentsByCode(c)
ans =
tContinent: [6x1 struct]
>> ans.tContinent
ans =
6x1 struct array with fields:
    sCode
    sName

```

Hodnoty sCode: AF, AM, AN, AS, EU, OC.

Hodnoty sName: Africa, The Americas, Antarctica, Asia, Europe, Oceania.

Metódy začínajúce sa slovom ‘List’ – vrátia tiež zoznam mien a ISO kódov všetkých krajín.

2.5 CurrencyConvertor

Webová služba *CurrencyConvertor* poskytuje informácie o aktuálnom menovom kurze na devízovom trhu. Táto služba má jednu funkciu s menom *ConversionRate*. Pri volaní *CurrencyConvertor* použitím metódy *ConversionRate* sa zadajú ISO kódy dvoch mien, ktorých vzťah nás zaujíma. WSDL služby *CurrencyConvertor* sa nachádza pod: <http://www.webservices.net/CurrencyConvertor.asmx?WSDL>.

Pr.:

```

>> cc=CurrencyConvertor;
>> ConversionRate(cc, 'EUR', 'SKK')
ans =
32.5137

```


Odpoveď je: 1 EUR sa rovná 32.5137 SKK.

2.6 CaptchaService

Mojím cieľom pri zavolaní tejto služby bolo dosiahnuť, aby sa stroje nestali používateľom stránky, totiž *CaptchaService* nám umožní vytvoriť obraz so slovom, ktoré v našom prípade slúži ako heslo vedúce k ďalšiemu použitiu služieb. Keďže vygenerované slovo stroje nedokážu prečítať, tým pádom nevedia postupovať ďalej. Pri tvorbe obrazu je potrebné zadať žiadené vlastnosti nasledovne:

```
vystup = generateCaptcha(captchaService, dĺžka_obrazu, šírka_obrazu,  
'rodina_písma', 'farba_písma', veľkosť_písma, zahmlenosť_pozadia,  
'farba_pozadia1', 'farba_pozadia2', kvalita, miera_skreslenia_písma,  
vzdialenosť_slova_na_Xosi, vzdialenosť_slova_na_Yosi);
```

Dĺžka a šírka obrazu, veľkosť písma, vzdialenosť začiatku slova na osi X a osi Y sa zadávajú v pixeloch. Farba písma, pozadia1 a 2 sa zadá pomocou anglického názvu farby. Zahmlenosť pozadia, kvalita a miera skreslenia písma sa zadefinujú číselnou hodnotou.

Napr.:

```
>> vystup = generateCaptcha (captchaService, 200, 200, 'Verdana',  
'black', 30, 0, 'green', 'blue', 100, 0, 10, 10)
```

Matlab vráti odpoveď:

```
vystup =
```

```
item: [2x1 struct]
```

Pole item zahŕňa v sebe URL pre vygenerovaný obraz a slovo, ktoré je na ňom zobrazené:

```
>>vystup.item(1)
```

```
ans =
```

```
key: 'CAPTCHAWORD'  
value: 'posited'
```

V tomto prípade slovo na obrázku je posited.

```
>>vystup.item(2)
```

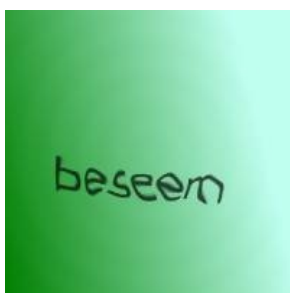
```
ans =
```

```
key: 'IMAGE'  
value: 'http://www.easycfm.com/forums/images/cotcha/35983.jpg'
```

Hore je uvedené URL vygenerovaného obrazu, ktorého zadáním do prehliadača dostaneme nasledovný výsledok:



V predchádzajúcom prípade som zahmlenosť pozadia definovala hodnotou 0. Pri použití nejakej väčšej hodnoty napr.: pre 10 výsledok vyzerá nasledovne:



V druhom prípade je pozadie zakalené oproti prvému.

Ďalej som volila hodnotu 5 pre mieru skreslenia písma, a dostala som obraz:



Tu sa dá vidieť väčšia miera skreslenia na rozdiel od ostatných prípadov.

Medzičasom služba *CaptchService* prestala fungovať ako služba na voľné použitie. Z tohto dôvodu som hľadala novú službu pre generovanie *captcha*, ktorá je použitá tiež v skripte v prílohe k práci.

2.7 TextCasing

Služba *TextCasing* slúži na zmenenie určitého slova, resp. určitých slov na žiadaný tvar rôznymi metódami. WSDL dokument služby *TextCasing* sa nachádza pod URL: <http://www.dataaccess.com/webservicesserver/textcasing.wso?WSDL>.

2.7.1 Funkcie TextCasing

AllLowercaseWithToken – premení všetky písmená slova na malé, pritom pridá nami predurčený znak pred a za každé písmeno. Príklad:

```
>> AllLowercaseWithToken(textcasing, 'RETAZEC', ' ')
ans =
r e t a z e c
```

V tomto prípade predurčeným znakom bola medzera.

AllUppercaseWithToken – premení všetky písmená slova na veľké, pred a za každé písmeno pridá žiadaný znak, v tomto prípade čiaru.

```
>> AllLowercaseWithToken(textcasing, 'retazec', ',')
ans =
,R,E,T,A,Z,E,C,
```

InvertCaseFirstAdjustStringToCurrent – premení prvé písmeno na opačnú veľkosť (veľké na malé a opačne) a veľkosť ostatných písmen prispôsobí k súčasnému stavu prvého písmena.

```
>> InvertCaseFirstAdjustStringToCurrent(TextCasing, 'Retazec')
ans =
retazec
```

InvertCaseFirstAdjustStringToPrevious – premení prvé písmeno na opačnú veľkosť (veľké na malé a opačne) a veľkosť ostatných písmen prispôsobí k stavu prvého písmena pred premenou.

```
>> InvertCaseFirstAdjustStringToPrevious(TextCasing, 'retazec')
ans =
Retazec
```

InvertStringCase – premení malé na veľké písmená a opačne :

```
>> InvertStringCase (textcasing, 'ReTazeC')
ans =
rEtAZEc
```

LowercaseWordsWithToken – táto funkcia slúži na premenu písmen viacerých slov, pritom možno zadať určitý znak, ktorý chceme pridať medzi slová.

```
>> LowercaseWordsWithToken (textcasing, 'RETAZEC1 RETAZEC2', ' ')\nans =\nretazec1  retazec2
```

TitleCaseWordsWithToken – táto funkcia je určená tiež pre viac slov, premení prvé písmeno slov na veľké.

```
>> TitleCaseWordsWithToken (textcasing, 'RETAZEC1 RETAZEC2', '_')\nans =\nRetazec1_Retazec2
```

UppercaseWordsWithToken – premení všetky písmená viac slov na veľké, možnosťou zadania určitého znaku medzi slová.

```
>> UppercaseWordsWithToken(textcasing, 'retazec1 retazec2', ' ')\nans =\nRETAZEC1  RETAZEC2
```

3. POUŽITIE VIACERÝCH SLUŽIEB

3.1 Vzájomné prepojenie služieb vo forme aplikácie

Jedným z cieľov okrem charakterizovania sieťových služieb bolo vytvoriť skript v Matlabe, ktorý by mal analyzovať správu za účelom zistenia menových kurzov. Službami, ktoré sa majú použiť v tomto skripte sú služby opísané v predchádzajúcej časti a tie sú: *CaptchaService*, *Calais*, *CountryInfoService*, *CurrencyConvertor* a *TextCasing*.

Postup po pustení plánovaného skriptu má byť nasledovný:

1. Užívateľ vloží text a slovo vygenerované *CaptchaService*om do príslušného formulára.
2. Analýzou textu pomocou služby *Calais* sa vytvorí zoznam krajín vystupujúcich v zadanom texte.
3. Pomocou *CountryInfoService* sa vytvorí zoznam mien a vlajok daných krajín.
4. Pomocou *CurrencyConvertor* sa vytvorí zoznam menových kurzov.
5. Vygeneruje sa tabuľka vlajkami reprezentujúcimi jednotlivé krajiny a menovými kurzami.
6. Užívateľ zadá sumu na prepočet.
7. Vygeneruje sa tabuľka prepočítanými hodnotami menových kurzov.

Pre fungovanie celého procesu som potrebovala napísať tri funkcie. Prvá funkcia s názvom *start* najprv zavolá službu *CaptchaService*.

V ďalších bolo potrebné vyzdvihnúť vygenerované slovo z výstupu *CaptchaService* a priradiť k premennej *slovo*, ktoré som zadefinovala ako globálnu premennú, aby sa dalo k jej hodnote pristupovať aj v iných funkciách:

```
function start(sprava, captcha)
global slovo
createClassFromWSDL('http://www.easycfm.com/webservices/captcha.cfc?wsdl');
g = captchaService;
captcha_out = generateCaptcha (g, 100, 100, 'Verdana', 'black', 30, 0
'purple','red',100, 0, 50, 50);
slovo = captcha_out.item(1).value;
```

Ďalej som z výstupu vyzdvihla URL vygenerovaného obrázku na ďalšie použitie:

```
obraz = captcha_out.item(2).value;
```

Nasledujúcim krokom bolo vytvoriť HTML dokument s názvom *home*. Tvorba takého dokumentu sa v Matlabe uskutoční zápisom HTML-kódu do súboru. HTML dokument sa pozostáva z hlavy, ktorá je ohraničená tagmi `<head></head>` a tela, ktoré je ohraničené tagmi `<body></body>`:

```
fl = fopen('home.html', 'w');
fprintf(fl, '<html>\n');
fprintf(fl, '<head></head>\n');
fprintf(fl, '<body><form name="formular1">\n');
fprintf(fl, 'Text:<textarea name="sprava" rows="10" cols="50"></textarea>\n');
fprintf(fl, '\n', captcha_out.item(2).value);
fprintf(fl, 'Captcha: <input name="captcha" type="text">\n');
fprintf(fl, '<input type="button" onclick="akcial()" value="Spracuj"');
fprintf(fl, '</form></body></html>\n');
fclose(fl);
web('home.html')
```

V tele som zadefinovala formulárové prvky typu textového poľa pre správu a heslo, ktoré zadá užívateľ a typu button, ktorého stlačením sa spracujú zadané údaje zavolaním funkcie *akcial* napísanej v hlavičke dokumentu v *javascripte*:

```
fprintf(fl, '<head><title>Štart</title><script type="text/javascript">\n');
fprintf(fl, 'function akcial() {\n');
fprintf(fl, 'area = new String(document.formular1.sprava.value);\n');
fprintf(fl, 'area = area.replace(/\\r|\\n|\\r\\n/g, " ");\n');
fprintf(fl, 'area = area.replace(/\\\'/g, "\'");\n');
fprintf(fl, 'commandURI="matlab:analyzuj('' + area;\n');
fprintf(fl, 'commandURI=commandURI+''', '' +
document.formular1.captcha.value;\n');
fprintf(fl, 'commandURI=commandURI+'''');";\n');
fprintf(fl, 'document.location=commandURI;\n');
fprintf(fl, '</script></head>\n');
```

Funkcia *akcial* odkazuje na M-file s názvom *analyzuj*, ktorému pošle zadanú správu a slovo na spracovanie po odoslaní formulára užívateľom. M-file *analyzuj* obsahuje funkciu *analyzuj*.

Funkcia *analyzuj* najprv porovná text zadaný do poľa s názvom *captcha*, čiže s textom priradeným ku globálnej premennej *slovo*. Ak sa nerovnajú, program sa ukončí a v Matlabe sa vypíše error s textom “Ľutujem, heslo je nesprávne“:

```
function analizuj(sprava, captcha)
global slovo
```

```

if ~isequal(captcha, slovo)
    error('Lutujem, heslo je nesprávne');

```

Ak sa rovnajú, funkcia pokračuje spracovaním správy. Na spracovanie správy sa zavolá služba *Calais*, ktorá nájde v texte mená krajín:

```

else
createClassFromWsd('http://api.opencalais.com/enlighten/?wsdl');
c =calais;
key = '6dber2gs3rp75xt52pspetrs';
params = '<c:params xmlns:c="http://s.opencalais.com/1/pred/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"><c:processingDirectives c:contentType="text/txt"
c:outputFormat="xml/rdf"></c:processingDirectives><c:userDirectives
c:allowDistribution="true" c:allowSearch="true" c:externalID="17cabs901"
c:submitter="ABC"></c:userDirectives><c:externalMetadata></c:externalMetadata>
</c:params>';
out = Enlighten(c, key, sprava, params);

```

Ďalším krokom by mal byť volanie služby *CountryInfoService*, k čomu je potrebné vyzdvihnúť z odpovedi *Calaisu* mená krajín, ďalej je potrebné zjednotiť mená krajín, aby sa k ďalšiemu spracovaniu nepoužilo názov tej istej krajiny viackrát. Predchádzajúce problémy vyrieši nasledovná časť skriptu:

```

krajiny = {};
pos = strfind(out, '<!--Country: ');
for i = pos
    start = i + length('<!--Country: ');
    krajina = '';
    for j = start:start+100
        if out(j) == '-'
            break
        else
            krajina = [krajina out(j)];
        end
    end
    krajiny{end+1} = krajina;
end
unikatne_krajiny = unique(krajiny);

```

Výsledkom je zoznam krajín, o ktorých sa hovorí v danej správe. Tu sa vyskytol taký problém, že krajiny, ktoré sa objavili v texte viackrát a pritom boli napísané rôznou veľkosťou písmen (napr. Slovakia a SLOVAKIA), tie bral Matlab ako dve rôzne slová.

Preto som zavolała ďalšiu službu *TextCasing* a zvolila som metódu *TitleCaseWordsWithToken*, ktorá premení prvé písmeno slov na veľké a ostatné na malé, pritom môžeme zadať určitý znak, ktorý chceme pridať medzi slová. To je veľkou výhodou v našom prípade, lebo mená niektorých krajín pozostávajú z viacerých slov, napr. Czech Republic. Riešenie pre predošlý problém je nasledovné:

```
createClassFromWsd('http://www.dataaccess.com/webservicesserver/textcasing.ws
o?WSDL');
t=TextCasing;
unikatne_krajiny1 = {};
for i = 1:length(krajiny)
    retazec_krajina = krajiny{i};
    unikatna_krajina = TitleCaseWordsWithToken(t, retazec_krajina, ' ');
    unikatne_krajiny1{end+1} = unikatna_krajina;
end
unikatne_krajiny2 = unique(unikatne_krajiny1);
```

Po riešení tohto problému som sa vrátila ku službe *CountryInfoService*, pomocou ktorej som dostala URL obrazov vlajok krajín, a meny použité v tých krajinách:

```
zoznam_meny = {};
zoznam_vlajok = {};
for i = 1:length(unikatne_krajiny2)
    jedna_krajina = unikatne_krajiny2{i};
    kod_krajiny = CountryISOCode(d, jedna_krajina);
    ci = FullCountryInfo(d, kod_krajiny);
    kod = ci.sCurrencyISOCode;
    vlajka = ci.sCountryFlag;
    zoznam_meny{end+1} = kod;
    zoznam_vlajok{end+1} = vlajka;
end
```

Okrem metód *CurrencyISOCode* a *CountryFlag* som použila aj *CountryISOCode* na zistenie ISO kódu krajiny. Totiž pri volaní ostatných metód je potrebné zadať ISO kód krajiny.

Výsledkom je zoznam mien, používaných v daných krajinách, ako i zoznam URL obrazov vlajok. Vo funkcii *analyzuj* poslednou použitou webovou službou je *CurrencyConvertor*, čiže ďalším krokom bolo volanie tejto služby nasledovným spôsobom:

```
createClassFromWsd('http://www.webserviceex.net/CurrencyConvertor.asmx?WSDL');
f = CurrencyConvertor;
zoznam_rozd = {};
for i = 1:length(zoznam_meny)
```



```

        mena1 = zoznam_meny{i};
        for j = 1:length(zoznam_meny)
            mena2 = zoznam_meny{j};
            rozd = ConversionRate(f, mena1, mena2);
            rozd = str2num(rozd);
            if rozd == 0
                rozd = 1;
            end
            zoznam_rozd{end+1} = rozd;
        end
    end
end

```

Výsledkom je zoznam menových kurzov. Posledným krokom v tejto funkcii je tvorba HTML dokumentu, ktorý zobrazuje tabuľku s vlajkami a výsledkami zo služby *CurrencyConvertor*. Túto úlohu splní nasledujúca časť skriptu:

```

fd = fopen('tabulka.html', 'w');
fprintf(fd, '<html><head><title>Tabulka mien</title></head>\n');
fprintf(fd, '<body><table border="1" cellspacing="2" cellpadding="2">\n');
fprintf(fd, '<tr><td></td>\n');
for i = 1:length(zoznam_vlajok)
    fprintf(fd, '<td>\n', zoznam_vlajok{i});
    fprintf(fd, '</td>\n');
end
fprintf(fd, '</tr>\n');
k = 1;
m = length(zoznam_meny);
for i = 1:length(zoznam_vlajok)
    fprintf(fd, '<tr><td>\n', zoznam_vlajok{i});
    fprintf(fd, '</td>\n');
    for j = k:m
        fprintf(fd, '<td>%f\n', zoznam_rozd{j});
        fprintf(fd, '</td>\n');
    end
    fprintf(fd, '<td><form name="formular%d"><input name="suma"
type="text" />\n', i);
    fprintf(fd, '<input type="button" onclick="akcia%d()" value="Spracuj"
></form>\n', i);
    fprintf(fd, '</td></tr>\n');
    k = k+length(zoznam_meny);
    m = k+length(zoznam_meny)-1;
end
fprintf(fd, '</table></html>\n');
fclose(fd);
web('tabulka.html')

```

Do tabuľky som pridala stĺpec s formulárom typu text, kde užívateľ môže zadať sumu, ktorá sa stlačením *spracuj* prepočítava na hodnoty ostatných mien. Formulárový prvok *spracuj* zavolá funkciu, ktorá je napísaná v hlavičke dokumentu v skriptovacom jazyku *javascript*. Hlavička dokumentu vyzerá nasledovne:

```
fprintf(fd, '<head><title>Calling MATLAB from JavaScript 3</title><script
type="text/javascript">\n');
for i=1:length(zoznam_vlajok)
    fprintf(fd, 'function akcia%d() {\n', i);
    fprintf(fd, 'commandURI="matlab:vypocitaj('" + document.formular
%d.suma.value;', i);
    fprintf(fd, 'commandURI+="'", '%d')";\n', i);
    fprintf(fd, 'document.location=commandURI;\n');
    fprintf(fd, '}\n');
end
fprintf(fd, '</script></head>\n');
```

Táto funkcia odkazuje na M-file *vypocitaj*, v ktorej sme najprv zadefinovali premenné pre funkciu *vypocitaj*, potom globálne premenné *zoznam_meny*, *zoznam_rozd*, *zoznam_vlajok*:

```
function vypocitaj(suma, index)
global zoznam_meny
global zoznam_rozd
global zoznam_vlajok
```

Výpočet vykoná nasledovný príkazový riadok:

```
suma = str2num(suma);
index = str2num(index);
nd = sqrt(length(zoznam_rozd));
tabulka = reshape([zoznam_rozd{:}], nd, nd)';
```

Tu sa najskôr prekonvertuje reťazec na číslo pomocou príkazu *str2num* a následne sa vynásobia hodnoty daného riadku hodnotou zadanou do formulára na konci príslušného riadku.

```
vysledok = tabulka(index, :)*suma;
```

Posledným krokom vo funkcii *vypocitaj* je tvorba HTML dokumentu ako v predchádzajúcich prípadoch:

```
fn = fopen('tabulka3.html', 'w');
fprintf(fn, '<html><head><title>Tabulka mien</title></head>\n');
fprintf(fn, '<body><table border="2" cellspacing="2" cellpadding="2">\n');
fprintf(fn, '<tr><td></td>\n');
for i = 1:length(zoznam_vlajok)
    fprintf(fn, '<td>\n', zoznam_vlajok{i});
    fprintf(fn, '</td>\n');
```

```


end
fprintf(fn, '</tr>\n');
fprintf(fn, '<tr><td>\n', zoznam_vlajok{index});
fprintf(fn, '</td>\n');
fprintf(fn, '<td>%f\n', vysledok);
fprintf(fn, '</td>\n');
fprintf(fn, '</tr>\n');
fprintf(fn, '</table></html>\n');
fclose(fn);
web('tabulka3.html')

```

3.1.1 Stránka v Matlabe

Spustením skriptu obsahujúceho funkciu *start* sa vytvorí nasledujúca stránka:

Text:



Captcha:

Do textového poľa s názvom *text* sa zadá správa, ktorú chceme analyzovať, do políčka s názvom *captcha* zadáme slovo, ktoré vidíme na obrázku.

Text:

meeting of foreign affairs ministers of NATO-member countries in Brussels, Kubis said that Slovakia may follow the lead of the neighbouring Czech Republic and sign a Memorandum of Understanding with the U.S.A. in late April, adding that further steps will depend on the results of negotiations in the upcoming weeks. Kubis once again declared that Slovakia is sticking to the joint steps called for by Brussels but wants to make use of scope for individual talks with the US.

chassid

Captcha:

chassid

Spracuj

Stlačením *spracuj* dostaneme takýto výsledok:

				
	1.000000	1.240100	0.062800	<input type="text"/> Spracuj
	0.806400	1.000000	0.050600	<input type="text"/> Spracuj
	15.926000	19.745000	1.000000	<input type="text"/> Spracuj

Ako vidíme, výsledkom je tabuľka, kde v prvom riadku a stĺpci sú zobrazené vlajky krajín, ktoré sú spomenuté v zadanej správe. Jednotlivé vlajky reprezentujú krajiny, resp. ich meny. Pri vyjadrení vzťahu medzi menami základom sú v riadku uvedené krajiny, to znamená, že v druhom riadku sú uvedené hodnoty mien prepočítané z jednej



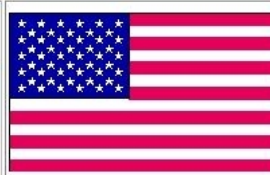
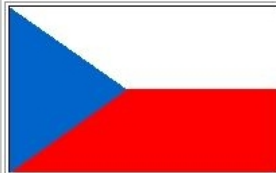
českej koruny na ostatné meny, v treťom riadku je základom jedna slovenská koruna, a v štvrtom riadku jeden americký dolár. Napr.:

2.riadok: 1CZK = 1CZK; 1CZK = 1,240100 SKK; 1CZK = 0,062800 USD.

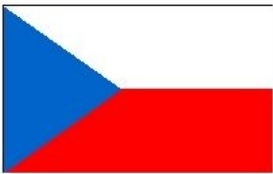

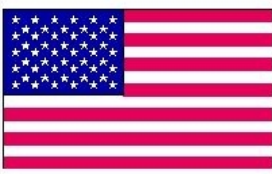

Na záver bol vytvorený formulár na konci každého riadku, do ktorých môžeme zadať sumu na prepočet:

				
	1.000000	1.237000	0.062900	<input type="text" value="2"/> <input type="button" value="Spracuj"/>
	0.808400	1.000000	0.050800	<input type="text" value="3"/> <input type="button" value="Spracuj"/>
	15.905000	19.674000	1.000000	<input type="text" value="4"/> <input type="button" value="Spracuj"/>

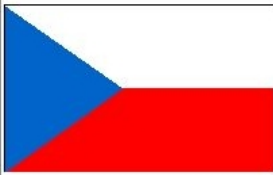



Formulár v druhom riadku slúži na prepočet zadanej sumy v českých korunách na ostatné meny. Zadaním napr.: 2 a stlačením tlačidla *spracuj* dostaneme nasledujúci výsledok:

			
	2.000000	2.474000	0.125800

Formulár v treťom riadku slúži na prepočet zadanej sumy v slovenských korunách na ostatné meny. Zadaním napr.: 3 a stlačením tlačidla *spracuj* dostaneme nasledujúci výsledok:

			
	2.425200	3.000000	0.152400

Formulár v štvrtom riadku slúži na prepočet zadanej sumy v amerických dolároch na ostatné meny. Zadaním napr.: 4 a stlačením tlačidla *spracuj* dostaneme nasledujúci výsledok:

			
	63.620000	78.696000	4.000000

ZÁVER

Cieľom bakalárskej práce bolo oboznámenie sa s WSDL službami a zobrazenie ich využitia v Matlabe. Na preukázanie rôznych možností využitia webových služieb v Matlabe som napísala skript, v ktorom sa použije viac na seba vzájomne nadväzujúcich služieb. Úlohou tohto matlabovského skriptu je vytvoriť stránku s formulárom, do ktorého zadaním anglickej novinovej správy a jej poslaním sa daná správa spracováva tak, aby sa vygenerovala tabuľka vlajkami a menovými kurzami v správe sa nachádzajúcich krajín. Pre uskutočnenie tohoto plánu najprv bolo potrebné nájsť vhodné služby na sieti. Rozhodla som sa použiť služby: *Calais*, *CaptchaService*, *CountryInfoService*, *CurrencyConvertor* a *TextCasing*. Služba *Calais* bola východiskom pre celý proces, analyzovala zadanú anglickú správu, a našla krajiny vyskytujúce sa v zadanej správe. Služba *CountryInfoService* vystupovala v programe ako druhá použitá služba, poskytovala informácie o používaných menách a o URL vlajok daných krajín. Ďalšia služba, ktorá hrala tiež dôležitú úlohu, bola *CurrencyConvertor*, ktorá poskytovala informácie o aktuálnom stave menových kurzov na trhu. Službu *TextCasing* som použila na korigovanie jedného vyskytnutého problému. Keď som mala k dispozícii všetky potrebné sieťové služby, skombinovala som ich do matlabovského skriptu tak, aby sa vytvorila hore spomenutá tabuľka s vlajkami a menovými kurzami.

Nakoniec sa podarilo vytvoriť plánovaný skript tak, aby sa s jeho spustením dosiahol požadovaný výsledok.

V budúcnosti by mohli byť webové služby využité častejšie aj v Matlabe, ich použitie totiž rozšíri jeho možnosti využitia. Výhodou webových služieb je, že ich použitie je jednoduché a ich vhodným skombinovaním do skriptu v Matlabe môžeme riešiť aj komplikovanejšie úlohy, na ktoré sme predtým nemali vhodné softwarové prostriedky.

LITERATÚRA

1. Martin Kuba: Webové služby,
(http://www.softec.sk/images/Softecon2007_Kuba.pdf)
2. WSDL – Web Services Description Language, UDDI – Universal Description,
Discovery and Integration, SOAP – Simple Object Access Protocol,
(<http://www.ekobit.sk>)
3. Jiří Kosek: Lehký úvod do XML, (http://www.cstug.cz/slt/01/plne_texty/13.pdf)
4. XML databázy a web services,
(http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/getfile.php/xml_databazy_a_web_services.doc)
5. (http://ics.upjs.sk/~rkb/web/s.ics.upjs.sk/_gabris/sps/)
6. Martin Kuba: Webové služby: WSDL: Struktura WSDL dokumentů,
(<http://www.ics.muni.cz/zpravodaj/articles/269.html>)