

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## TRANSFORMACE WWW STRÁNEK NA WEBOVÉ SLUŽBY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

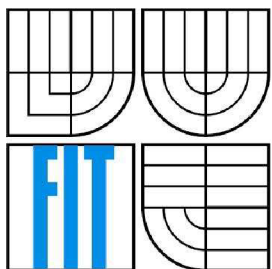
AUTHOR

JAN KLEMENT

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# TRANSFORMACE WWW STRÁNEK NA WEBOVÉ SLUŽBY

TRANSFORMATION WWW PAGES INTO WEB SERVICES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN KLEMENT

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR WEISS

BRNO 2008

## **Abstrakt**

Předmětem této práce bylo seznámit se s webovými aplikacemi, pochopit jejich architekturu a způsob komunikace a srovnat použití webových aplikací s použitím webových služeb. Webové služby umožňují hromadné zasílání a zpracování dat a částečnou automatizaci webu. Cílem práce bylo navržení aplikace, která umožní obecnou transformaci WWW stránky na webovou službu, a její implementace. Aplikace by měla sloužit pro demonstrační účely.

## **Klíčová slova**

Web, WWW stránka, HTTP, HTML, PHP, informační systém, webová služba, XML, WSDL, komunikační protokol, SOAP, registr služeb, UDDI.

## **Abstract**

The objective of this thesis was to acquaint oneself with web applications, study their architecture and communication and compare using of web applications and web services. Web services enable mass data sending and partially automate web applications. The aim of the thesis was to design a transformation of WWW page into web service in general. Furthermore, an application providing the transformation was implemented.

## **Keywords**

Web, WWW page, HTTP, HTML, PHP, information system, web service, XML, WSDL, communication protocol, SOAP, service registry, UDDI.

## **Citace**

Jan Klement: Transformace WWW stránek na webové služby, bakalářská práce, Brno, FIT VUT v Brně, rok 2008

# Transformace WWW stránek na webové služby

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Weisse  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jméno Příjmení  
Datum

## Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Petru Weissovi za kvalitní podklady a odbornou pomoc během celé tvorby práce.

© Jan Klement, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah .....	1
Seznam zkratek .....	3
1 Úvod.....	5
1.1 Cíl práce .....	5
1.2 Struktura práce .....	6
2 Webové technologie.....	7
2.1 WWW stránky.....	7
2.1.1 Hypertext Transfer Protocol.....	7
2.1.2 HTML, XHTML .....	8
2.1.3 Informační systém.....	10
2.2 Webové služby .....	11
2.2.1 Extensible Markup Language .....	12
2.2.2 WSDL .....	14
2.2.3 SOAP .....	16
2.2.4 UDDI.....	17
3 Transformace webových stránek na webové služby .....	19
3.1 Analýza WWW stránek.....	19
3.1.1 Formulář.....	19
3.1.2 Stav stránky.....	19
3.1.3 Zdrojový kód.....	20
3.2 Transformace na webové služby .....	21
3.2.1 Využití služby WWW stránky .....	22
3.2.2 Využití webové služby .....	22
3.3 Návrh aplikace.....	23
3.3.1 Inicializace služby - startup.....	23
3.3.2 Vykonávání služby – server .....	24
3.3.3 Žadatel o službu – client .....	24
4 Implementace .....	25
4.1 startup.....	25
4.2 server .....	25
4.3 client.....	26
4.4 Pozitiva a negativa .....	26

4.4.1	Výhody poskytnutého řešení.....	26
4.4.2	Nevýhody .....	27
4.5	Problémy při vývoji.....	27
5	Závěr .....	29
	Literatura .....	31
	Seznam příloh .....	32

# Seznam zkratek

Pro snazší orientaci v textu uvádím seznam použitých zkratek, jejich význam a funkci, kterou představují.

- **CGI** (Common Gateway Interface ) je protokol pro propojení externích aplikací s webovým serverem.
- **DOM** (Document Object Model) je objektově orientovaná reprezentace XML nebo HTML dokumentu.
- **DTD** (Document Type Definition) je jazyk pro popis struktury XML případně SGML dokumentu.
- **FTP** (File Transfer Protocol) je protokol aplikační vrstvy z rodiny TCP/IP, je určen pro přenos souborů mezi počítači, na kterých mohou běžet velmi rozdílné operační systémy.
- **GPL** (General Public License) licence pro svobodný software.
- **HTML** (HyperText Markup Language) je značkovací jazyk pro hypertextové dokumenty.
- **HTTP** (Hypertext Transfer Protocol) je internetový protokol určený původně pro výměnu hypertextových dokumentů ve formátu HTML.
- **HTTPS** (Hypertext Transfer Protocol Security) je nadstavba počítačového protokolu HTTP, která poskytuje zvýšenou bezpečnost před odposloucháváním či podvržením dat.
- **MySQL** je databázový systém, vytvořený švédskou firmou MySQL AB.
- **PHP** (PHP: Hypertext Preprocessor) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek.
- **RFC** (Request For Comments) používá se pro označení řady standardů a dalších dokumentů popisujících Internetové protokoly, systémy apod.
- **SAX** (Simple API for XML) umožňuje sériový přístup ke XML.
- **SGML** (Standard Generalized Markup Language) je univerzální značkovací metajazyk, který umožňuje definovat značkovací jazyky jako své vlastní podmnožiny.
- **SMTP** (Simple Mail Transfer Protocol) je internetový protokol určený pro přenos zpráv elektronické pošty.
- **SOAP** (Simple Object Access Protocol) je protokolem pro výměnu zpráv založených na XML přes síť.
- **SSL** (Secure Sockets Layer) je vrstva, která poskytuje zabezpečení komunikace šifrováním a autentizací komunikujících stran.
- **TCP** (Transmission Control Protocol) je jedním ze základních protokolů sady protokolů Internetu, konkrétně představuje transportní vrstvu.

- **UDDI** (Universal Description, Discovery and Integration) je platformově nezávislá specifikace pro vyhledávání a registraci webových služeb, založená na XML.
- **URL** (Uniform Resource Locator) je řetězec znaků s definovanou strukturou sloužící k přesné specifikaci umístění zdrojů informací na Internetu.
- **XML** (eXtensible Markup Language) je obecný značkovací jazyk.
- **XSLT** (eXtensible Stylesheet Language Transformations) slouží k převodům zdrojových dat ve formátu XML do libovolného jiného požadovaného formátu, nejčastěji HTML, jiného XML nebo libovolných jiných datových struktur.
- **XHTML** (eXtensible Hypertext Markup Language) je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW.
- **WS** (Web Service, ) - webová služba je řešení, jak spolu mohou komunikovat a vyměňovat si informace aplikace přes Internet.
- **WSDL** (Web Services Description Language) popisuje co nabízí webová služba za funkce a způsob, jak se jí na to zeptat.
- **WWW** (World Wide Web, zkráceně web) je označení pro aplikace internetového protokolu HTTP. Je tím myšlena soustava propojených hypertextových dokumentů.



# 1 Úvod

Používání internetu se každým dnem zvyšuje a internet jako takový je v současnosti nejpoužívanějším zdrojem informací. S jeho rozšířením roste poptávka po službách, které lze přes web uskutečnit. Lidé nakupují v on-line obchodech, rezervují si pobyty v hotelech, zamlouvají si prostory k pronájmu, zjišťují stavy svých účtů a kont, objednávají si služby a zboží. Všechny tyto služby již mají na internetu svou reálnou podobu a skutečně jich lze využít přímo z pohodlí domova.

Webové aplikace, které umožňují tyto služby, prostřednictvím WWW stránek uživateli postupně sdělují informace o žádané službě a vyžadují prostřednictvím webového rozhraní vstupní informace pro vykonání služby. Tyto informace se vyplňují do formulářů, které jsou odesílány a na webovém serveru zpracovány. Data jsou postupně zadávána do jednotlivých polí formuláře.

To s sebou přináší skutečnosti. Data nelze zasílat hromadně a celý proces musí obsluhovat osoba sedící u počítače.

Webová služba umožňuje komunikaci a výměnu informací mezi aplikacemi, přes Internet. Byla navrhnutá a je používána pro strojovou výměnu informací po síti. Transformací webové aplikace na webovou službu dosáhneme tedy nejen jisté míry automatizace výměny informací přes Internet, ale i možnosti hromadného zadávání dat.

Výhodami webové služby bude kratší doba potřebná k realizaci služby a definovaný formát pro zasílání a příjem dat.

## 1.1 Cíl práce

Cílem práce je navrhnout postup řešení pro obecnou transformaci WWW stránky na webovou službu a posléze implementovat aplikaci, která tuto přeměnu umožní.

Dílejší cíle:

- seznámit čtenáře s prostředky k vytváření webových aplikací a služeb,
- provést srovnání konceptu webové aplikace a webové služby z různých pohledů (implementační, aplikační, atd.)
- analyzovat webové stránky vzhledem k její struktuře a obsahu,
- navrhnout obecný postup pro provedení automatické transformace WWW stránky na webovou službu,
- implementovat aplikaci, která bude transformaci demonstrovat,
- provést ukázkou činnosti zvoleného postupu na příkladu,
- vytvořit manuál k používání demonstrační aplikace.

## 1.2 Struktura práce

Z hlediska metodického přístupu je práce členěna do tří celků.

První z nich se zabývá problematikou webových technologií. Přibližuje způsob, jak pracují webové aplikace prostřednictvím webových stránek, seznamuje čtenáře s protokoly, jazyky a dalšími nástroji pro jejich tvorbu a vysvětluje pojmy HTTP, HTML, PHP a MySQL. Dále se věnuje webové službě. Popisuje, z jakých částí se skládá a jak probíhá její komunikace. Popisuje architekturu i způsob komunikace webových služeb a vysvětlí pojmy XML, SOAP, WSDL a UDDI.

V druhé části budeme seznámeni s analýzou WWW stránky vzhledem k jejímu obsahu. Zjistíme elementy, které ovlivňují obsah stránky. Na základě provedené analýzy bude proveden návrh aplikace, která bude umožňovat obecnou transformaci webové stránky na webovou službu. Nejdříve probereme inicializační část, která vytvoří rozhraní pro komunikaci WS, po té serverovou část, tedy poskytovatele WS a nakonec klientskou část, která využívá WS. Bude provedeno srovnání použití služby přes WWW stránku a přes WS, dozvíme se o výhodách i nevýhodách poskytnutého řešení. Budeme také srozuměni s problémy a překážkami ve vývoji aplikace.

Následovat bude část o implementaci aplikace pro transformaci. Podrobněji se zaměříme na jednotlivé části aplikace vyplynulé z návrhu.

Závěr celé práce bude obsahovat konečný stav projektu. Najdeme zde jeho zhodnocení, dosažené cíle, nedostatky a možná vylepšení pro další vývoj.

## 2 Webové technologie

V této kapitole bude čtenář uveden do rozsáhlé problematiky vytváření webových aplikací a webových služeb. Kapitola je členěna na dva dílčí celky.

První z nich popisuje služby a protokoly nutné k vytváření WWW stránek a webových aplikací. V této podkapitole bude čtenář seznámen s pojmy HTTP protokol, jazyk HTML (XHTML), informační systém a nástroji s ním spojenými – PHP, MySQL, které jsou pro pochopení problematiky v širších souvislostech nezbytné.

V pořadí druhá podkapitola se zabývá teoretickým přístupem k problematice webových služeb. Přibližuje význam pojmů XML, WDSL, SOAP a UDDI.

### 2.1 WWW stránky

Jak již bylo naznačeno, první kapitola bude věnována problematice vytváření WWW stránek. Jsou zde uvedeny nástroje pro tvorbu stránek, pro komunikaci mezi webovým serverem a klientem (internetovým prohlížečem). Pro názornost jsou připojeny ilustrace.

#### 2.1.1 Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) je internetový protokol původně určený pro výměnu tzv. hypertextových dokumentů. Dokumenty jsou psány v jazyku HTML (viz níže). Definice protokolu (aktuální verze 1.1) je definována v protokolu RFC 2616. V oblasti internetové sítě je tento protokol spolu s SMTP (Simple Mail Transfer Protocol, protokol určený pro přenos zpráv internetové pošty) jednoznačně nejpoužívanějším protokolem.

Nyní je používán i pro přenos dalších informací. Umí přenášet soubory různých typů, pomocí formátu XML umožňuje realizaci webových služeb, zpřístupňuje i další protokoly jako např. FTP (File Transport Protocol, protokol určený pro přenos souborů mezi počítači), nebo již zmiňovaný SMTP.

Protokol funguje způsobem klient – server. V této relaci se pod pojmem klient rozumí uživatelský počítač komunikující obvykle pomocí svého internetového prohlížeče. Pod označením server je myšlen web server posílající požadované objekty. Používá se bezpečného spojení TCP. Klient inicializuje TCP spojení (vytvoří socket) a posílá serveru textový dotaz obsahující označení požadovaného dokumentu, informace o klientovi (schopnosti prohlížeče), apod. Server akceptuje TCP spojení a odpovídá opět v textové formě. Odpověď obsahuje výsledek dotazu (např. zda se

podařilo hledaný soubor najít, jakého je typu) a za ním následují data žádaného dokumentu. Komunikace mezi klientem a serverem je postavena na sekvenci dotazů a odpovědí.

```
GET /wiki/Wikipedie HTTP/1.1
Host: cs.wikipedia.org
User-Agent: Mozilla/5.0 Gecko/20040803
Firefox/0.9.3
Accept-Charset: UTF-8,*
```

*Ilustrace 1: HTTP dotaz*

```
HTTP/1.0 200 OK
Date: Fri, 15 Oct 2004 08:20:25 GMT
Server: Apache/1.3.29 (Unix) PHP/4.3.8
X-Powered-By: PHP/4.3.8
Vary: Accept-Encoding, Cookie
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Content-Language: cs
Content-Type: text/html; charset=utf-8
```

*Ilustrace 2: HTTP odpověď*

HTTP je *bezstavový protokol* – server neukládá žádné informace o předešlých dotazech, neumí zachovat stav komunikace. Toto může způsobovat problémy s implementací složitých systému komunikujících přes HTTP. Proto byl protokol rozšířen o HTTP cookies – ty umožňují uchovávat informace o stavu připojení na klientském počítači.

HTTP používá některé další aplikace jako na příklad tzv. jednotný lokátor prostředků URL (Uniform Resource Locator), který jednoznačně specifikuje umístění zdroje v Internetu.

K protokolu HTTP existuje také jeho bezpečnější verze HTTPS. HTTPS přenášená data šifruje pomocí SSL (Secure Sockets Layer), což zabezpečuje celý přenos dat.

## 2.1.2 HTML, XHTML

HTML (HyperText Markup Language) je značkovací jazyk pro hypertextové dokumenty. V současné době jde o nejpoužívanější jazyk pro vytváření webových stránek. Jeho vývoj byl ovlivněn schopnostmi internetových prohlížečů, které zpětně ovlivňují definici jazyka.

HTML je typem dokumentu SGML (Standard Generalized Markup Language), kde je značkám přidělena sémantika hypertextového dokumentu. Přípravovaná verze 5.0 už ale závislá na SGML nebude.

Jazyk je charakterizován množinou značek (tagů) a jejich atributů (doplňující informace upřesňující vlastnosti elementu) definovaných pro danou verzi. Tagy se ohraničují úhlovými závorkami a mohou být párové (název koncového tagu je uvozen '\'), či nepárové. Prostor vymezený tagy určuje právě sémantiku elementů, ty se mohou do sebe vnořovat.

Každá verze má vlastní definici pravidel - DTD (Document Type Definition), která definuje pro danou verzi přípustné elementy a atributy.

Dokument může kromě textu a tagů obsahovat direktivy (začínají dvou-znakem <! a jsou určeny pro internetový prohlížeč, který dokument zpracovává), komentáře (texty, které slouží programátorovi pro orientaci a prohlížeč je ignoruje), kódy skriptovacích jazyků (zejména PHP, CGI, atd. viz kapitola 2.1.3) a definice událostí (JavaScript). Všechny tyto prostředky pro tvorbu webových stránek jsou neodmyslitelnou součástí informačních systémů (viz. níže).

Struktura dokumentu je jasně daná [1], HTML verze 4.01:

- Deklarace DTD – je povinná až ve verzi 4.01, je uvedena direktivou <!DOCTYPE.
- Kořenový element – element html (značky <html> a </html>) reprezentuje celý dokument. Kořenový element je povinný, ale otevírací a ukončovací značka samotná povinná není (pokud tyto značky nebudou v těle dokumentu uvedeny, prohlížeč si je sám doplní podle kontextu).
- Hlavička – obsahuje metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení. Hlavička je uzavřena mezi značky <head> a </head>. Element head je opět povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.
- Tělo dokumentu – obsahuje vlastní text dokumentu. Vymezuje se značkami <body> a </body>. Element body je povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.

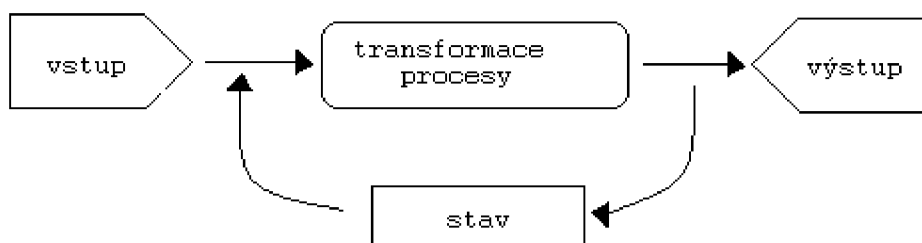
XHTML (eXtensible Hypertext Markup Language) – značkový jazyk pro tvorbu hypertextových dokumentů v prostředí WWW. Oproti HTML je tu však několik rozdílů. Tím největším rozdílem je nutnost ukončení tagů a to i nepárových. Dále pak názvy tagů musí být malými písmeny (XHTML je case-sensitive), hodnoty atributů musí být v uvozovkách. Další rozdíly jsou popsány v [2].

## 2.1.3 Informační systém

Informační systémy (IS) jsou systémy pro práci s daty. Jedná se především o jejich sbírání, údržbu a poskytování.

*Informacemi* míníme sdělení, které odstraňuje nejistotu a přináší jistou míru znalostí, *daty* míníme jakékoli zaznamenané poznatky či fakta. Jako zvláštní pojem zde vystupuje také *znalost* představující zobecnění poznání určité části reality. Informaci je možné také chápat jako data s přidaným významem.

Informační systém pracuje s konceptuálními (nehmotnými) zdroji. Skládá se ze vstupní a výstupní části, kudy proudí informace do resp. ze systému. Uvnitř systému probíhá transformace dat pomocí předem definovaných procesů. Důležitou součástí IS je zpětná vazba, využívající uloženého stavu systému. Schéma IS je zobrazeno na následující ilustraci (ilustrace 3). Více informací zde [3].



*Ilustrace 3: Schéma IS*

Okruh problémů IS je značně rozsáhlý, k naší problematice postačí implementace webových aplikací. Každá webová aplikace se skládá z několika webových stránek. Pro jejich tvorbu a provoz se často používá trojkombinace Apache server, jazyk PHP a databázový systémem MySQL.

PHP (PHP: Hypertext Preprocessor) skriptovací jazyk nezávislý na platformě a určený především pro programování dynamických webových stránek. Skripty zapsané v tomto jazyce mohou být začleněny přímo v kódu HTML (nebo XHTML), případně mohou být externě připojeny, což je vhodné pro tvorbu webových aplikací. PHP skripty jsou prováděny na straně serveru, klient obdrží až výsledek jejich provedení. Je oblíbeným nástrojem pro tvorbu webového prostředí hlavně díky jednoduchosti použití. Výhodou je i jeho syntaxe, která čerpá z populárních programovacích jazyků (C, Perl, Java) a nechává tak vývojáři poměrně velkou svobodu. Obsahuje velké množství knihoven pro zpracování grafiky, textu, práci se soubory, přístup k databázi a podporu řady internetových protokolů. Technologie je volně dostupná ke stažení na <http://www.php.net/downloads.php>.

MySQL je multiplatformní databázový systém komunikující prostřednictvím jazyka SQL (Structured Query Language). Aby bylo možné jej optimalizovat na vysokou rychlost, bylo nutné

provést zjednodušení jím poskytovaných služeb. Jeho předností je kromě rychlosti i jednoduchá implementace, která umožňuje snadnou realizaci na různých operačních systémech. MySQL je k dispozici pod bezplatnou licencí GPL na <http://dev.mysql.com/downloads/>

## 2.2 Webové služby

Webová služba (Web Service, WS) je podle definice W3C<sup>1</sup>[4] softwarový systém umožňující aplikacím komunikaci a výměnu informací přes internet. Společně fungují na bázi architektury klient – server, kde komunikace probíhá přes SOAP zprávy. Jazyk WSDL je nástrojem pro tvorbu rozhraní WS. UDDI registr uchovává seznam dostupných webových služeb. Veškerá komunikace probíhá pomocí platformě nezávislých XML dokumentů. Jednotlivé komponenty budou podrobněji rozebrány dále.

*„Webová služba je softwarový systém zkonstruovaný k podpoře interakce mezi stroji přes síť. Má rozhraní popsané ve strojově zpracovatelném formátu (specificky WSDL). Ostatní systémy interagují s webovou službou způsobem předepsaným jejím popisem za pomoci SOAP zpráv, typicky dopravovaných použitím HTTP s XML zápisem v součinnosti s ostatními webovými standardy.“*[2]

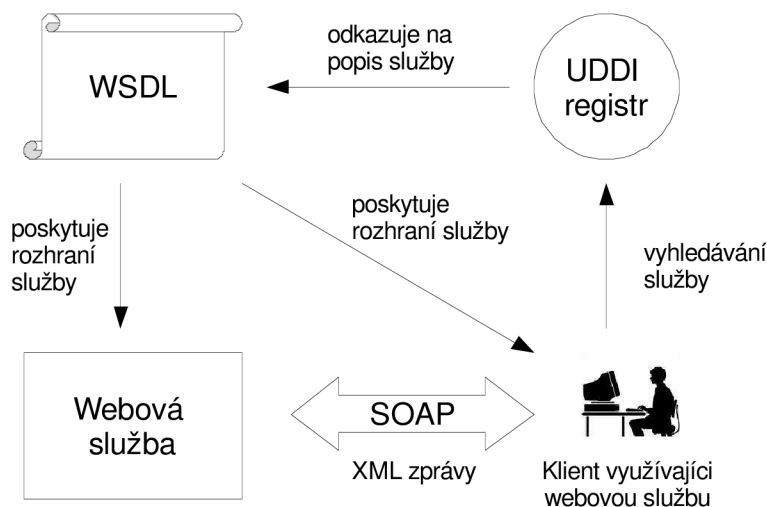
Výhodou webových služeb je nezávislost na platformě. Na klientské straně není třeba žádný speciální SW, stačí pouze podpora HTTP a XML. Na straně serveru je nutný webový server a engine podporující speciální programy běžící na straně serveru.

Webové služby jsou samopopisné. Definice a popis zprávy je posílán společně se samotnou zprávou. Webové služby k přenosu a transformaci dat využívají několik technologií založených na XML [5].

- **XML** (Extensible Markup Language) – základ, na kterém jsou webové služby postaveny. Poskytuje jazyk pro definici dat a popis jak s těmito daty nakládat. XML reprezentuje skupinu provázaných specifikací vydaných a spravovaných organizací W3C.
- **WSDL** (Web Services Description Language) – definuje rozhraní webových služeb, datové typy, typy zpráv a vzory protokolů.
- **SOAP** (Simple Object Access Protocol) je protokol pro přenos XML dokumentů přes síť.
- **UDDI** (Universal Description, Discovery, and Integration) – registr webových služeb a mechanismus vyhledávání používaný pro třídění informací a získávání odkazů na WS.

---

1 World Wide Web Consortium (W3C) je mezinárodní konsorcium jehož členové společně s veřejností vyvíjí webové standardy pro World Wide Web.



*Ilustrace 4: Architektura webových služeb*

### 2.2.1 Extensible Markup Language

Extensible Markup Language (XML) je obecný značkovací jazyk, umožňující definovat značkovací jazyky, je určen pro výměnu dat mezi aplikacemi a pro publikaci dokumentů. Je to základ, na kterém jsou webové služby postaveny. Určuje formát zpráv pro komunikaci (SOAP) i způsob, jakým jsou služby definovány (WSDL). Umožňuje popsat strukturu dokumentu z hlediska jeho věcného obsahu, nezabývá se vzhledem dokumentu.

XML je podobný jazyku HTML – obsahuje elementy, atributy a hodnoty, ale na rozdíl od HTML jich umožňuje definovat neomezený počet. Tvoří tak stromovou strukturu, kde uzly stromu tvoří tagy, listy mohou být tagy, atributy i hodnoty.

To však s sebou přináší také nevýhody. XML umožňuje definovat vlastní elementy a je proto obtížné zajistit, aby byly sémanticky vždy správně pochopeny. Tento problém řeší XML schémata, ta obsahují definice i sémantiku elementů. Pokud obě komunikující strany sdílí stejné XML schéma, mohou si být jisty, že jednotlivé elementy mají stejný význam [6].

Práci s XML dokumenty lze usnadnit pomocí aplikačních rozhraní. Mezi nejznámější patří DOM (Document Object Model) a SAX (Simple API for XML). DOM prochází dokument opakovaně po částech, zatímco SAX dokument prochází pouze jednou. SAX je proto rychlejší, vhodnější pro inicializaci aplikace z XML a méně náročný na paměť. Oba programy jsou volně dostupné a vhodné pro většinu platform. Další informace lze nalézt na internetu: <http://www.saxproject.org/> a <http://www.w3.org/DOM/>.



```

<?xml version="1.0" encoding="UTF-8"?>
<recept jméno="chleba" čas_přípravy="5 minut" čas_vaření="3 hodiny">
  <titulek>Jednoduchý chleba</titulek>
  <přísada množství="3" jednotka="šálky">Mouka</přísada>
  <přísada množství="0,25" jednotka="unce">Kvasnice</přísada>
  <přísada množství="1,5" jednotka="šálku">Horká voda</přísada>
  <přísada množství="1" jednotka="kávová lžička">Sůl</přísada>
  <instrukce>
    <krok>Smíchejte všechny přísady dohromady a dobře prohnětte.</krok>
    <krok>Zakryjte tkaninou a nechejte hodinu v teplé místnosti.</krok>
    <krok>Znovu prohnětte, umístěte na plech a pečte v troubě.</krok>
  </instrukce>
</recept>

```

*Ilustrace 5: Ukázka XML dokumentu, obrázek převzat z [5]*

### 2.2.1.1 Výhody XML

XML poskytuje přesně danou formu pro přenos dat. Jeho přednostmi jsou:

- **standardní formát** – odesílání i příjem dat je nezávislý na zpracovávající aplikaci i na operačním systému,
- **přizpůsobivost** – pro přidání nových dat stačí dodefinovat novou značku,
- **otevřenost** – definice je vytvořena konsorciem W3C a je volně přístupná.

### 2.2.1.2 Technologie využívající XML

XML pro svou univerzálnost pronikl do různých technologií. Níže uvedené pojmy úzce souvisí s problematikou webových služeb [9].

- **XML namespaces** – specifikace standardizovaná konsorciem W3C, která umožňuje rozlišení značek jazyků. Řeší problém unikátnosti elementů.
- **XML schéma** – vytváří pravidla struktury validních XML dokumentů, umožňuje definování vlastních typů a jmenných prostorů. Nabízí se srovnání s DTD. Oproti DTD je XML schéma mocnější, specifikováno opět v XML formátu a je snáze čitelné.
- **XSLT** - (Extensible Stylesheet Language Transformations) – jazyk pro převod mezi XML dokumenty, případně pro převod do jiných jazyků.
- **DOM** a **SAX** – aplikace pro práci s XML dokumenty.

```

<-- XML dokument -->
<zamestnanec uid="kroupaP">
  <jmeno>Petr</jmeno>
  <prijmeni>Kroupa</prijmeni>
  <oddeleni>Údržba</oddeleni>
  <plat>14700</plat>
</zamestnanec>

<-- XML schema -->
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="zamestnanec">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="jmeno" type="xs:string"></xs:element>
        <xs:element name="prijmeni" type="xs:string"></xs:element>
        <xs:element name="oddeleni" type="xs:string"></xs:element>
        <xs:element name="plat" type="xs:integer"></xs:element>
      </xs:sequence>
      <xs:attribute name="uid" type="xs:string"></xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

*Ilustrace 6: Ukázka XML schéma, obrázek převzat z [5]*

## 2.2.2 WSDL

WSDL (Web Services Description Language) je jazyk popisující webovou službu ve formátu XML. WSDL dokument předepisuje formát dat, které mohou být službě zasílány. Popisuje metody, které služba může využívat i způsob, jak je zavolat. Podporované operace a zprávy jsou popsány abstraktně a potom se omezují na konkrétní síťový protokol a formát zprávy. Specifikuje, kde je daná služba dostupná (např. HTTP, SMTP, URL) a určuje způsob navázání a průběhu komunikace. WSDL se proto často používá ve spojení se SOAP (Simple Object Access Protocol, viz níže) a popisuje rozhraní webové služby [6].

Klient i poskytovatel služby (server) musí mít přístup ke stejnému WSDL dokumentu, který je nutný pro vzájemné porozumění. Klient musí vědět, v jakém formátu bude zpráva posílána, a server musí vědět, jak příchozí zprávu zpracovat.

Celý WSDL dokument je rozčleněn do několika oddílů určující jeho popis. *Abstraktní popis* udržuje neměnnost a kompletnost (integritu) popisu služby. Popisuje její rozhraní nezávisle na operačním systému, technologii i způsobu komunikace. Jedná se o tyto oddíly: types, message, operation a port type.

*Konkrétní popis* aplikuje logiku popsanou abstraktním modelem na reálnou implementaci, navazuje komunikaci přes konkrétní protokol - oddíly binding, service a port.

Přehled základních oddílů:

- **types** – popis datových typů ve formě XML schémat,
- **message** – formát zprávy určené k přenosu,
- **portType** – abstraktní popis operací,
- **binding** – specifikuje použití konkrétního protokolu,
- **port** – specifikuje adresu pro spojení,
- **service** - používán ke spojení k příslušnému portu.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name=" "
  targetNamespace=" "
  xmlns:tns=" "
  ... >

<!-- definice typu -->
<types>
  <schema targetNamespace=" " ... >
    <element name=" " type=" "/>
    <element name=" " type=" "/>
  </schema>
</types>

<!-- zpravy -->
<message name=" Request">
  <part name=" " element=" "/>
</message>
<message name=" Response">
  <part name=" " element=" "/>
</message>

<!-- operace -->
<portType name=" ">
  <operation name=" ">
    <documentation>
    </documentation>
    <input message=" Request"/>
    <output message=" Response"/>
  </operation>
</portType>

<!-- adresy komunikacnich bodu -->
<service name=" ">
  <documentation> </documentation>
  <port name=" " binding=" ">
    <SOAP:address location=" "/>
  </port>
</service>
</definitions>
```

*Ilustrace 7: Schéma WSDL [5]*

## 2.2.3 SOAP

SOAP (Simple Object Access Protocol) je protokol pro výměnu zpráv zapsaných ve XML formátu. Pro přenos používá internetových protokolů HTTP (nebo HTTPS), díky tomu je platformě nezávislý. Jde o bezstavový protokol (podobně jako HTTP). Vývoj SOAP je zaměřen na jednoduchost a rozšiřitelnost. Název je poněkud zavádějící, protože SOAP není jednoduchý a ani neslouží k přístupu objektů. Často se používá jako náhrada za RPC (Remote procedure call) sloužícího ke vzdálenému volání procedur. Formát SOAP tvoří základní vrstvu komunikace mezi webovými službami a poskytuje prostředí pro tvorbu složitější komunikace.

SOAP byl původně navržen firmami Microsoft a IBM pro vzdálené volání procedur. Nyní je ve specifikaci 1.2 organizace W3C [7].

### 2.2.3.1 SOAP zpráva

Zpráva SOAP zapouzdřuje *hlavičku* (head) a *tělo* (body) přenášených dat do *obálky* (envelope), kterou tvoří obecná pravidla protokolu.

Hlavička je nepovinná. Obsahuje nutné informace pro všechny služby, které mohou zprávu obdržet. Určuje tedy způsob, jakým bude zpráva zpracována. Pro doplnění informací viz [6].

V těle se nacházejí data v XML formátu. Mohou se zde vyskytovat sekce pro chyby (faults) a informace pro jejich zpracování. Tělo zprávy obsahuje hlavní značku (obrázek 2.7), která má stejný název jako procedura, která má být volána. Její atribut *xmlns* určuje místo (URL), kde je služba popsána. Vnořené elementy jsou parametry volání. Zpráva přinášející odpověď (obrázek 2.8) v těle obsahuje opět jednu hlavní značku, jejíž název je složen z názvu volané procedury a řetězce „Response“. Vnořené elementy obsahují návratové hodnoty [8].

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Header/>
  <env:Body>
    <jeStatEU xmlns="urn:URIslužby">
      <stat xsi:type="xsd:string">Polsko</stat>
    </jeStatEU>
  </env:Body>
</env:Envelope>
```

*Ilustrace 8: SOAP dotaz*

```

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <env:Body>
    <jeStatEUResponse xmlns="urn:URIsluzby">
      <odpoved xsi:type="xsd:boolean">true</odpoved>
    </jeStatEUResponse>
  </env:Body>
</env:Envelope>

```

*Ilustrace 9: SOAP odpověď*

Je-li třeba posílat data, která nelze popsat pomocí XML (např. obrázky, zvuky), je nutno je poslat jako *přílohy* (attachments).

Zapisování dat do formátu XML má své klady i zápory. Zápis je poměrně jasně čitelný pro uživatele (osobu), ale ne pro počítač. Ten musí dokument složitě parsovat, což je náročné na operační paměť a čas procesoru. Existují systémy využívající pro komunikaci binární zápis XML (například CORBA nebo GIOP). Ten je pro člověka nečitelný. Vývoj počítačů jde ale rychle dopředu a časová prodleva nebo nároky na operační paměť již v současné době nejsou problémem.

## 2.2.4 UDDI

UDDI (Universal Description, Discovery and Integration) je platformě nezávislá specifikace pro vyhledávání a registraci webových služeb, založená na XML [8]. Není to nic jiného než další webová služba, která udržuje „seznam“ webových služeb popsaných pomocí WSDL spolu s jejich autory nebo poskytovateli. Je reprezentováno XML dokumentem a podporuje přenos dat přes SOAP. Webovou službu lze pochopitelně poskytovat i bez jejího zaregistrování v UDDI, ale pak bude daná služba známá a přístupná pouze těm, kterým to bude sděleno. Snaha o přístup k webové službě je však většinou opačná [8, 10].

Registrovat se lze u jednoho z hlavních poskytovatelů UDDI. Informace jsou uloženy do databáze poskytovatele a postupem času kopírovány do ostatních databází.

Informace v UDDI jsou členěny do tří skupin. Bílé stránky obsahují popis společnosti nabízející službu (kontakt, adresa, apod.). Žluté stránky zahrnují průmyslové kategorie založené na standardních systematikách. Zelené stránky obsahují technické informace o uvedených službách.

V UDDI existuje několik způsobů, jak vyhledávat webové služby (ilustrace 10). Vyhledávat lze podle různých kritérií (klíčová slova, názvy firem, obory, místo, apod.). Záleží na specifikaci webové služby.

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <get_businessDetail generic="2.0" xmlns="urn:uddiorg:api_v2">
      <businessKey="E99023-4142-9A08-9DE3-D239808119A0">
    </get_businessDetail>
  </Body>
</Envelope>
```

*Ilustrace 10: Dotaz do UDDI [5]*

# 3 Transformace webových stránek na webové služby

V této kapitole se budu zabývat vlastní prací. Mým úkolem bylo analyzovat WWW stránky vzhledem k jejich struktuře a obsahu. Na základě toho jsem navrhl možné řešení obecné transformace WWW stránek na webové služby. Výsledkem návrhu bude aplikace, která umožní jednotnou výměnu dat mezi klientem a serverem pomocí SOAP zpráv.

## 3.1 Analýza WWW stránek

Pojďme se podívat na dynamiku webových stránek. Vstupní atributy, které ovlivňují, jak se bude daná stránka chovat, je možno zadávat do URL spolu s adresou stránky. Atributy se zadávají ve tvaru 'název=hodnota' a oddělují se znakem '&' (odesílání atributů metodou GET).

Je možno požit i metodu POST, která své parametry obsahuje přímo v těle požadavku, takže hodnota parametrů není v URL vidět. Obsah webové stránky se změní prakticky jen kliknutím na odkaz nebo odesláním formuláře, což odpovídá použití metod GET a POST. Zabýváme se změnou obsahu z pohledu uživatele. Nebereme proto v potaz automatické přesměrování stránky, změny stránky vyvolané JavaScriptem (probíhají jen na straně klienta) a další.

### 3.1.1 Formulář

Formulář je posílán metodami GET nebo POST. Při odeslání metodou GET se z formuláře stává odkaz, prvky formuláře slouží jako atributy. Každý prvek formuláře má své jméno, proto je možné přiřadit konkrétnímu jménu konkrétní hodnotu. Zpracování dat z formuláře obstará PHP skript (viz kapitola 2.1.3) uvedený jako atribut *action* tagu `<form>` v HTML kódu (viz ilustrace 12).

### 3.1.2 Stav stránky

Stav aplikace je tedy dán aktuálním vykonáním příslušného skriptu, vyplněním a odesláním formuláře, nebo zobrazením stránky při kliknutí na odkaz. Webové aplikace si částečně dokáží udržet stav pro konkrétního uživatele pomocí cookies. Cookies je malé množství dat, která server pošle prohlížeči. Ten je uloží na počítači uživatele. Při každé další návštěvě téhož serveru pak prohlížeč tato data posílá serveru zpět.

### 3.1.3 Zdrojový kód

Nyní budeme zkoumat elementy WWW stránky přímo ve zdrojovém kódu. HTML (XHTML) kód je stromová struktura s kořenovým elementem `<html>`. Aby bylo možné ji zkoumat, dokument musí být validní. Ke kontrole a opravě zdrojového kódu stránky je možno použít volně dostupný Open Source nástroj HTML Tidy implementovaný v populárních programovacích jazycích (C, Java, Perl), dostupný na stránkách <http://tidy.sourceforge.net/>. HTML Tidy dokáže rozčlenit strukturu kódu, analyzovat ji a do jisté míry i opravit. Dokáže doplnit chybějící elementy, opravit chybné zanořování prvků, strukturovat kód graficky a hlavně transformovat HTML kód na striktnější XHTML. O své činnosti a o zjištěných chybách podává informace do speciálního textového souboru.

Z hlediska možnosti interakce s webovou aplikací nás zajímá jen určitá část kódu stránky. Zaměříme se konkrétně na ilustraci 11. Je na ní zobrazena WWW stránka umožňující vyhledávání informací o vysokých školách. Skládá se z několika částí:

- hlavička stránky – nadpis a obrázek,
- odkazy ovlivňující obsah stránky uvedený pod nimi, odkazují na služby, které webová stránka poskytuje,
- formulář pro vyhledávání informací podle kombinací parametrů zadávaných v polích formuláře.

**Poradna o vysokých školách**

[vyhledat info o VŠ](#)   [vyhledat info o fakultách](#)   [vyhledat info o přijímacích řízeních](#)   [přihlásit se](#)

**Vyhledávání informací o fakultách**

název fakulty:

zkratka VŠ:

zkratka fakulty:

typ VŠ:

kraj:

město:

Ilustrace 11: WWW stránka umožňující vyhledávání informace o vysokých školách



Podstatnou částí z našeho pohledu je právě formulář uvedený ve spodní polovině stránky, jehož odesláním (kliknutím na tlačítko 'vyhledat') se vykoná skript obsluhující formulář a zobrazí se požadované informace – změní se stav aplikace (obsah stránky).

Změnu stránky tedy umožňuje jen několik elementů. Tato část kódu je zvýrazněna na ilustraci 12.

```

<div id="hlavicka">
  <h1>Poradna o vysokých školách </h1></div>
  <div id="panel">
    <table><tr>
      <td><a href="/findVS.php">vyhledat info o VŠ</a></td>
      <td><a href="/findF.php">vyhledat info o fakultách</a></td>
      <td><a href="/resultP.php">vyhledat info o přijímacích řízeních</a></td>
      <td><a href="/login.php"><font color="green">přihlásit se</font></a></td>
    </tr>
  </table></div><br><br><br>
<h2>Vyhledávání informací o fakultách</h2>
<form action="/resultF.php" method=post>
  <table border=0, width="30%" align="center">
    <tr><td>název fakulty: </td><td><input name="nazev" type="text"></td></tr>
    <tr><td>zkratka VŠ: </td><td><input name="zkratka_vs" type="text"></td></tr>
    <tr><td>zkratka fakulty: </td><td><input name="zkr_f" type="text"></td></tr>
    <tr><td>typ VŠ</td><td><select name="typ">
      <option value="vsechno_t">všechny typy VŠ</option>
      ...
      <option value="technicka">technická</option></select></td></tr>
    <tr><td>kraj</td><td><select name="kraj">
      <option value="vsechno_k">všechny kraje</option>
      ...
      <option value="vysocina">Vysočina</option></select></td></tr>
    <tr><td>město <td><input name="mesto" type="text"></td></tr>
  </table>
  <input name="vyhledat" type="submit" value="vyhledat"></form>
```

*Ilustrace 12: HTML kód stránky uvedené na ilustraci 11*

## 3.2 Transformace na webové služby

Po úspěšné analýze webové stránky je třeba projít celý strom HTML tagů a najít v něm podstatné informace pro transformaci. Nejpodstatnější částí budou formuláře představující logiku dynamických stránek. Data vyplněná do formulářů jsou odeslána na server a zpracována pomocí skriptů.

Transformace WWW stránek na webové služby nebude jednoduchá. Služby poskytující webové stránky mohou být různé a postaveny na rozdílných principech. Bude nutné zvolit velmi sofistikované řešení, aby bylo možno transformaci provést obecně.

Než přistoupíme k řešení samotné transformace na webové služby, ukážeme si, jak vypadá používání služeb přes WWW stránky, jak vypadá použití webové služby a jaké jsou praktické rozdíly mezi oběma variantami.

### 3.2.1 Využití služby WWW stránky

Nejdříve si popíšeme, jak to vypadá, když užíváme služeb webové stránky přímo přes webové rozhraní. Na prohlížeči si otevřeme příslušnou WWW stránku, která poskytuje službu. Po shlédnutí několika úvodních informací se dostaneme do stavu, kdy po nás budou vyžadována konkrétní vstupní data (viz např. ilustrace 11). Postupně vyplníme všechna povinná pole formuláře a nakonec jej odešleme stisknutím příslušného tlačítka. Odesláním formuláře se na serveru aktivuje příslušný PHP skript (např. resultF.php – viz ilustrace 11), který zpracuje odeslaná data (naváže komunikaci s databází, zpracuje data, aktualizuje soubory, apod.) a zobrazí na stránce výsledek své činnosti.

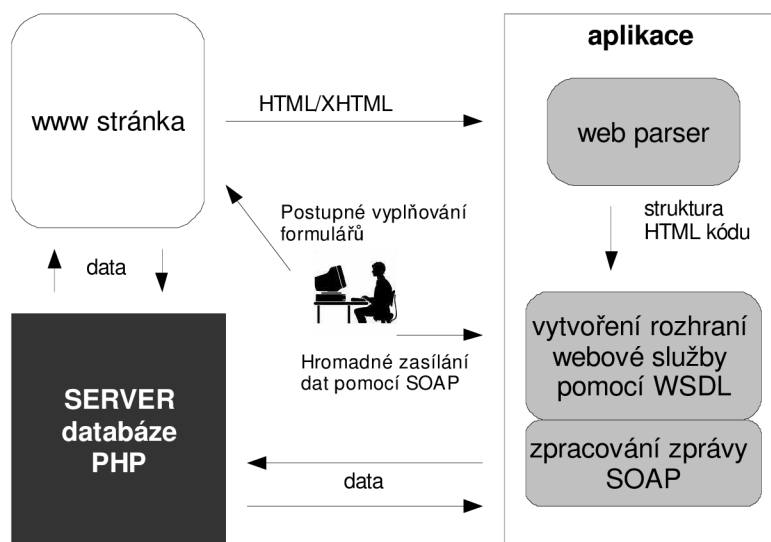
Je nutné vyplnit každé pole formuláře zvlášť, snadno dochází k překlepům na klávesnici a celou proceduru je často nutné opakovat znova. Tato metoda využití služby po internetu navíc vyžaduje personální obsazení u počítače. To vše zabírá poměrně hodně času.

### 3.2.2 Využití webové služby

Použití webové služby se bude zdát na první pohled jednodušší a elegantnější. Klient vyhledá webovou službu v registru UDDI (viz ilustrace 4) a vytvoří SOAP dotaz, v němž budou začleněna odesílaná data. Server, na němž je služba provozována, zprávu přijme, obsažená data zpracuje (stejným způsobem jako se zpracovávají data z formulářů webové stránky, tj. za pomoci stejného PHP skriptu) a na závěr pošle zpět odpověď opět ve formě SOAP zprávy.

Použití webové služby bude nezávisle na použitém prohlížeči a bude bezesporu rychlejší. Data se navíc dají zapsat konzistentně a poslat najednou. Výhodou je, že klient i server mají k dispozici stejný dokument WSDL definující způsob jejich komunikace, takže obsluha klienta může probíhat strojově. To napomáhá automatizaci a šetří pracovní sílu.

Nebude to ale jednoduché. K tomu, abychom mohli používat konkrétní webovou službu transformovanou z stránky, potřebujeme aplikaci, která provede transformaci webové stránky na webovou službu. Webovou službu je tedy potřeba ručně inicializovat. Tuto problematiku zobrazuje ilustrace 13, kde je znázorněna komunikace mezi uživatelem a serverem poskytujícím službu prostřednictvím své WWW stránky. Uživatel může buď postupně vyplňovat formulář, nebo může pomocí vytvořené aplikace posílat hromadně data přímo serveru. Zpracování dat a výsledek operací je u obou variant shodný, webová služba by měla umět poskytnout minimálně to, co umožňuje webová aplikace.



Ilustrace 13: Schéma komunikace mezi uživatelem a serverem

### 3.3 Návrh aplikace

K obecné transformaci WWW stránky na webovou službu bude potřeba vytvořit aplikaci, která postupně provede analýzu webové stránky, definuje rozhraní pro komunikaci prostřednictvím WSDL dokumentu, zaregistruje se do adresáře UDDI a bude schopna jak přijímat, tak odesílat SOAP zprávy.

Je jasné, že tento problém bude potřeba rozdělit na několik částí:

- **startup** – iniciátor webové služby, používá se jen při prvním použití,
- **server** – čeká na příjem SOAP požadavků od klienta, zpracovává je, vypisuje zprávy o činnosti, která probíhá mezi serverem a klientem (program client viz níže) a odesílá žádaná data zpět v očekávaném formátu, určeným WSDL dokumentem,
- **client** – zasílá požadavky serveru, přijímá odpovědi.

Aplikace bude komunikovat se serverem prostřednictvím HTTP protokolu a SOAP zpráv.

Nyní přejdeme ke konkrétním částem aplikace.

#### 3.3.1 Inicializace služby - startup

Program startup naváže komunikaci s příslušným serverem a stáhne na počítač WWW stránku, kterou chceme transformovat. Dále proběhne analýza stránky, na jejímž základě se vytvoří WSDL dokument.

Bude-li potřeba, můžeme použít nástroj HTML Tidy, abychom zjistili validitu HTML kódu. Je-li kód validní a elementy jsou řádně ukončené, můžeme přejít k další fázi. Je třeba projít kód a vybrat z něj ty části, které jsou důležité pro vytvoření WSDL dokumentu. Podstatné bude nalézt formuláře a prvky potřebné pro zobrazení výsledků operací. Naopak přeskočit můžeme nepodstatné elementy jako jsou hlavičky a patky dokumentů, nadpisy, obrázky a jiné grafické elementy (viz vyznačené části HTML kódu na ilustraci 12). Na základě tohoto torza HTML kódu webové stránky vytvoříme WSDL dokument, čímž definujeme popis rozhraní, přes které budeme komunikovat.

WSDL dokument je potřeba ručně umístit na konkrétní místo na internetu, odkud bude volně ke stažení.

Nyní je potřeba službu zaregistrovat do UDDI registru. To bude provedeno pomocí zaslání SOAP zprávy s údaji o poskytované službě a o poskytovateli služby na server provozovatele UDDI. Server odpoví na registraci SOAP zprávou s přiřazenými klíči.

Tím je inicializace hotova.

### **3.3.2 Vykonávání služby – server**

Server je program, který realizuje službu. Po spuštění čeká na SOAP požadavky klientů. Má k dispozici WSDL dokument umístěný na webu, takže ví, jakým způsobem má reagovat na příchozí SOAP zprávy.

Po obdržení SOAP zprávy provede požadovanou akci, odešle zpět klientovi žádaná data a čeká na další požadavky. Během své činnosti server vypisuje na obrazovku průběh zpracování.

Program bude běžet nepřetržitě po celou dobu poskytování služby. Server by měl být konkurentní, tzn. měl by být schopen přijímat více dotazů najednou.

### **3.3.3 Žadatel o službu – client**

Program client je aplikace, která umožňuje komunikaci se serverem. Nejdříve vyhledá službu v registru UDDI zasláním SOAP zprávy v předepsaném formátu. Jako odpověď přijdou informace o hledané službě, URL serveru, na němž je služba poskytována a URL dokumentu WSDL, který službu popisuje.

Aplikace kontaktuje server, na kterém je služba poskytována, naváže spojení a zašle SOAP zprávu. Zpráva je opět zaslána ve formátu určeným přístupným WSDL dokumentem.

Čeká na příchozí SOAP zprávu s výsledky činnosti webové služby. Pro přijetí zprávy budou zobrazena příchozí data a činnost programu je ukončena. Program client slouží pouze pro demonstraci funkčnosti serveru.

## 4 Implementace

V této kapitole popíši postup, jakým jsem postupoval při implementaci. Samotná implementace není jádrem práce, protože se týká především vytvoření demonstrační aplikace.

Aplikace je naprogramována v jazyce C/C++. Celý systém je vytvořený ve vývojovém prostředí Code::Blocks, testován a překládán na OS Linux.

### 4.1 startup

Slouží jako inicializační část webové služby. Program při spuštění očekává dva povinné parametry - URL stránky, která bude transformována a URL, kam bude uložen WSDL dokument. Rozdělí URL adresu na server a cestu ke stránce. Síťová komunikace se provádí pomocí BSD soketů. Následuje spojení s webovým serverem. Program žádá od serveru poslání dokumentu umístěného na uživatelském zadaném URL. Soubor je stažen a uložen do počítače jako textový soubor.

Dále proběhne analýza stáhnutého souboru s HTML kódem. Soubor je načten jako stream a zpracovává se po řádcích. Vyhledává element `<form>`. Jestliže HTML kód neobsahuje element `<form>`, aplikace se ukončí s chybou. V opačném případě zkoumá text ohraničený tagy `<form>` a `</form>`. Hledá prvky `<input>`, `<select>`, `<option>` a `<textarea>`. Zkoumá jejich atributy `name` a `type`. Na základě výskytu jednotlivých elementů vytváří soubor `wsdl.xml`, což je dokument WSDL popisující rozhraní webové služby.

Umístění WSDL dokumentu je dáno druhým parametrem při spuštění programu. Soubor `wsdl.xml` se na server musí umístit ručně po vykonání programu.

Program pokračuje v činnosti. Nyní registruje webovou službu na pevně daný server poskytující UDDI registr pomocí zaslání SOAP zprávy. Jako odpověď přijde stejná zpráva s vyplněnými klíči. Tím program končí.

Nyní je potřeba ručně umístit soubor `wsdl.xml` na URL zadané v druhém vstupním parametru programu.

### 4.2 server

Program očekává jeden povinný parametr – URL umístění WSDL dokumentu `wsdl.xml`. Běží v nekonečné smyčce a očekává příjem SOAP zprávy od klienta. Na standardní výstup vypisuje zprávy o své aktuální činnosti.

Jakmile obdrží zprávu od klienta, posílá nazpět připravenou SOAP zprávu. Na základní úrovni je výměna dat implementována pomocí BSD socketů.

Program server je nutno spustit na počítači, který službu poskytuje.

## 4.3 client

Program očekává dva povinné parametry – název SOAP zprávy, která bude odesílána (případně cestu k souboru v počítači) a klíčové slovo, které bude použito k vyhledání webové služby. Program odešle SOAP zprávu na server spravující UDDI registr. Zpět obdrží SOAP zprávu obsahující URL WSDL dokumentu popisující webovou službu a URL serveru, na němž je webová služba poskytována.

WSDL dokument definuje, jak má vypadat zpráva pro komunikaci se serverem. Klient dokument analyzuje a nalezne klíčové informace pro vytvoření SOAP zprávy, kterými jsou název služby a vstupní parametry. Poté klient sestaví SOAP zprávu a odesílá ji pomocí BSD socketů serveru. Dále čeká na odpověď a na závěr ukončí spojení.

## 4.4 Pozitiva a negativa

Ve své práci řeším obecný problém transformace webových stránek na webové služby. V následující kapitole uvedu, jaké jsou výhody a nevýhody tohoto řešení.

### 4.4.1 Výhody poskytnutého řešení

Rád bych pro přehlednost uvedl výhody využití webové služby transformované z WWW stránky oproti použití původní webové aplikace.

- Velikou výhodou je nesporně snaha o automatizaci z hlediska uživatele a rychlejšího přístupu a zpracování žádaných informací ze serveru. Nebude potřeba lidské síly pro vyhledávání a postupné vypisování formulářů na webu.
- Další výhodou je možnost hromadného zadávání dat podle předem přesně stanoveného formátu. Práce se tím zrychlí a uživatel opravdu dostane data, o která žádal, navíc v očekávaném formátu. Není třeba posouvat kurzor myši na stránce k vyplnění dalších polí formuláře.
- Program může běžet na pozadí, stačí pouze poskytnout SOAP zprávu klientovi. Výsledky jsou uloženy na počítači klienta.
- Není nutno zabývat se informačními stránkami o správném vyplňování formulářů. Vše je přesně specifikováno v dostupném WSDL souboru.

- K použití webové služby není potřeba internetový prohlížeč, uživatel se nemusí zabírat jejich optimalizací.

## 4.4.2 Nevýhody

Nové řešení problému s sebou pochopitelně přináší nevýhody, které většinou plynou z nezvyklého používání služeb. Je třeba se seznámit s problematikou webových služeb.

- Nevýhodou je poměrně složitá manipulace se SOAP zprávami, které se musí dopředu generovat. Je nutné proniknout do problematiky XML a SOAP a důkladně se seznámit s prostředky pro práci s dokumenty.
- Další nevýhodou oproti ručnímu vypisování WWW stránky je inicializační proces, který musí proběhnout při zavedení služby proběhnout. Můžou vzniknout komplikace při pokusech o kontaktování serverů poskytujících služby i serverů poskytujících UDDI registr.
- Všechny webové stránky nejsou validní. Je to sice chyba vývojářů, ale i s těmito chybami je potřeba počítat. Pro uskutečnění transformace je potřeba nástroj HTML Tidy (viz kapitola 4.3.1).
- Prostředí není implementováno v grafickém režimu. Pro mnoho uživatelů může toto představovat nepřekonatelný problém.

## 4.5 Problémy při vývoji

Během vývoje produktu jsem se setkal s několika problémy. Jejich uvedením se snažím o omezení a duplikaci případných chyb pro případné následovníky své práce.

Asi největším problémem byl původní návrh aplikace. Původně jsem aplikaci navrhl jako jeden ucelený program, který měl obsluhovat veškerou činnost při transformaci webové stránky na webovou aplikaci. Aplikace měla provádět akce postupně a měla čekat na vstupy uživatele. Tento návrh byl špatný z mnoha důvodů – ztratily se výhody použití tohoto přístupu k využití služby poskytované přes internet (automatizace, obsluha strojem, konkurentost), program byl nepřehledný a často z důvodů špatně navržených algoritmů kolaboval.

Součástí programu měly být i soubory programu HTML Tidy, které měly HTML kód stažené webové stránky kontrolovat a opravovat. Samotné HTML Tidy však představovalo obrovské procento z celé aplikace a samotné jádro řešení prakticky úplně zakrylo.

Dlouho jsem váhal, který programovací jazyk použít. Dlouho se nabízel jazyk Java, který má k dané problematice rozumně zpracované rozsáhlé knihovny. K programování aplikace v tomto jazyce jsem se ale neodvážil, protože se nepovažuji se za člověka znalého tento jazyk natolik, abych v něm program vyvíjel.

Implementace nakonec proběhla v jazyce C/C++, což díky popularitě jazyka poskytuje prostor pro případné rozšíření aplikace.

Jedním z největších problémů je nedostatek literatury o této problematice. V češtině podstatě neexistuje použitelná publikace poskytující základní informace a uvedení do problému. V angličtině je jich málo. Doufám, že tato práce částečně vyplní volný prostor na knižním trhu a poskytne zájemcům z řad široké veřejnosti základní informace v ucelené podobě.



## 5 Závěr

Cílem práce bylo navrhnout postup řešení pro obecnou transformaci WWW stránky na webovou službu a posléze implementovat nástroj, který tuto přeměnu provede.

V teoretické části této práce jsme se seznámili s webovými technologiemi. Předvedli jsme si způsob komunikace mezi serverem a klientem při používání WWW stránek, seznámili jsme se s prostředky, které při komunikaci vystupují. Uvedli jsme si komunikaci přes HTTP, strukturu WWW stránky v HTML kódu a vysvětlili jsme si, na jakém principu je postaven informační systém a jakých technologií využívá.

Dále jsme se seznámili s webovými službami. Byl vysvětlen pojem XML a jeho stěžejní postavení při užívání WS. Zjistili jsme, jak se definuje rozhraní webových služeb pomocí WSDL dokumentů a popsali jsme si způsob komunikace mezi aplikacemi pomocí SOAP zpráv. Bylo vysvětleno použití UDDI registru, uchovávajícího informace o webových službách a jejich poskytovatelích. Z jejich popisu můžeme pozorovat univerzálnost webových služeb, jejich potenciál a široké uplatnění při komunikaci přes Internet.

V další kapitole byla provedena analýza WWW stránky vzhledem k její struktuře popsané jazykem HTML. Na základě analýzy bylo navrženo řešení pro obecnou transformaci WWW stránky na webovou službu. V téže kapitole je poskytnuto srovnání mezi použitím webových aplikací a používáním webových služeb, jsou uvedeny výhody i nevýhody použití poskytovaného řešení.

Ke zvolení tématu této práce mě vedlo několik faktorů. Webové služby jsou zajímavou kapitolou využívání Internetu pro komunikaci mezi aplikacemi. Problém obecné transformace webové stránky na webovou službu, pokud vím, zatím nebyl řešen a představoval pro mne výzvu.

Práce bude sloužit hlavně pro demonstraci řešení problému a bude používána ke studijním účelům. Při stahování obecné WWW stránky z internetu je totiž stažen pouze HTML kód bez skriptů, které vykonávají službu na webovém serveru. Není proto možné správným způsobem zpracovat vstupní data. Tento problém se jednoduše vyřeší zakomponováním skriptu do serverové části aplikace, který dodá vývojář webové aplikace. Serverová část v současné verzi vypisuje pouze textové zprávy o průběhu své činnosti a posílá na zpět předchystanou SOAP zprávu. Dalším omezením je fakt, že při analýze webové stránky neznáme způsob zobrazení výsledku operace. Data mohou být zobrazena v úplně rozdílném formuláři, tabulce, či dokonce v textové formě. Proto je WSDL dokument zatím navržen tak, aby odpověď serveru měla jen jednu návratovou hodnotu informující o provedení služby (true/false). Zde je možno na práci navázat analýzou formátu zobrazení výstupních dat webové aplikace a na jejím základě upravit návratové hodnoty služby ve WSDL dokumentu.

Konečná podoba celého řešení umožňuje propojení s pracemi zaměřenými na tvorbu a poskytování UDDI registru a obecně na práce zaměřené na používání webových služeb.

Díky této práci mi byl objasněn princip webových služeb, který odhalil jejich velký potenciál při komunikaci přes Internet. Seznámil jsem se s řadou pro mě neznámých technologií a doplnil jsem si znalosti z oblasti problematiky webu. Byl mi poskytnut prostor pro nalezení a řešení nového způsobu využívání služeb přes Internet. Práce též prohloubila mé programovací zvyky a schopnosti.

Při vypracovávání projektu jsem vyhledal a nastudoval poměrně značné množství informací, které jsem čerpal převážně z Internetu a z aktuálních knih a publikací napsaných v angličtině. Na internetu je značné množství informací o problematice, často jsem se však setkal s tím, že některé údaje byly neaktuální nebo nepřesné.

Osobně ve webových službách vidím velkou budoucnost. Jsem rád, že jsem získal znalosti a dovednosti z tohoto oboru.

# Literatura

- [1] W3C. *HTML 4.01 Specification*. Working group note, 1999, [online], [cit. duben 2008].  
URL <http://www.w3.org/TR/html401/>
- [2] W3C. *XHTML™ 1.1 - Module-based XHTML*. Working group note, 2001, [online], [cit. duben 2008].  
URL <http://www.w3.org/TR/2001/REC-xhtml11-20010531/>
- [3] Hruška, T.; Křivka, Z.: *Informční systémy (IIS, PIS)*. Studijní opora, Vysoké učení technické, Brno, 2007.
- [4] OASIS. *Reference Model for Service Oriented Architecture*, Committee Draft 1.0, [online], [cit. květen 2008].  
URL <http://www.oasis-open.org/committees/download.php/16587/wd-soa-rm-cd1ED.pdf>
- [5] Kubát, D.: *Rozhraní webových služeb*. Bakalářská práce, Vysoké učení technické, Brno, 2007.
- [6] Papazoglou, M., P.: *Web Services: Principles and Technology*. Pearson Education, 2008, ISBN 978-0-321-15555-9.
- [7] W3C. *Simple Object Access Protocol (SOAP) 1.1*. Working group note, 2000, [online], [cit. duben 2008].  
URL <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [8] Erl, T.: *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, 6. vydání, 2006. ISBN 0-13-185858-0.
- [9] Kuba, M.: *Web services*. [online][cit. květen 2008]. URL  
[http://www.ics.muni.cz/~makub/soap/MartinKuba\\_WebServices\\_Datakon2006\\_clanek.pdf](http://www.ics.muni.cz/~makub/soap/MartinKuba_WebServices_Datakon2006_clanek.pdf)
- [10] Newcomer, E.: *Understanding Web Services: XML, WSDL SOAP and UDDI*. Addison-Wesley, 2002, ISBN 0-201-75081-3.

# Seznam příloh

Příloha 1. CD se zdrojovými texty a manuálem k aplikaci