

# **Historie a vývoj webových technologií z pohledu vývojáře**

The history and the development of web technologies from the  
perspective of a web developer

Dušan Ház

---

Bakalářská práce  
2008



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

\*\*\* nescannované zadání str. 1 \*\*

## **ABSTRAKT**

Pohled do historie, současnosti a vývoje Internetu. Analýza webových prohlížečů, standardizace při tvorbě webových stránek. Nové trendy ve vývoji webových prohlížečů a webdesignu.

Klíčová slova:

Internet, Arpanet, Javascript, Cookies, Blog, Webové inženýrství

## **ABSTRACT**

Look to the history, presents and development internet. Analysis web browsers, standardization at production web pages. New trends under development web browsers and web - design.

Keywords:

Internet, Arpanet, Javascript, Cookie, Blog, Web engineering

Motto:

...moderní informační technologie budou sloužit nejenom pro doplnění a vylepšení starých pracovních postupů a obchodních modelů. Naopak, tyto technologie podnítí vznik zcela nových typů služeb a tím také příležitostí pro šikovné dodavatele. Dokonce bych řekl, že do dvaceti let bude trh se službami vypadat zcela jinak. Jak? To předem nevíme, neboť je to otázka přirozeného vývoje, jaký nelze zcela naplánovat. Jediné můžeme s jistotou předpovědět – bude podstatně uživatelsky přívětivější, transparentnější a mnohem více zákaznický orientovaný...

Profesor Richard Susskind, Office of Lord Chief Justice, Londýn, V. Británie  
(Oxford University Press 2003)

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....  
Podpis diplomanta

**OBSAH**

<b>ÚVOD.....</b>	<b>8</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 HISTORIE INTERNETU .....</b>	<b>11</b>
1.1 USA.....	11
1.2 EVROPA.....	13
1.3 ČESKÁ REPUBLIKA .....	13
1.4 PROJEKT TEN - 34 CZ .....	16
1.5 PROJEKT QUANTUM (TEN-155) .....	17
1.6 OSTATNÍ SÍTĚ NA ÚZEMÍ ČESKÉ REPUBLIKY .....	18
1.7 PEERING - SPÁSA DOMÁCÍHO INTERNETU .....	20
<b>2 HISTORIE WEBOVÝCH STANDARDŮ .....</b>	<b>22</b>
2.1 PROČ DODRŽOVAT WEBOVÉ STANDARDY .....	24
2.2 VÝHODY WEBOVÝCH STANDARDŮ .....	26
2.3 DODRŽOVÁNÍ WEBOVÝCH STANDARDŮ V PRAXI .....	26
<b>3 HISTORIE A VÝVOJ WEBOVÝCH PROHLÍŽEČŮ.....</b>	<b>27</b>
3.1 CO TO VLASTNĚ JE? .....	27
3.2 HISTORIE .....	27
<b>4 VÝVOJ ASP A PHP – CESTA SPRÁVNÝM SMĚREM?.....</b>	<b>30</b>
4.1 ASP.....	30
4.1.1 Jak to celé funguje.....	30
4.1.2 Výhody asp.net oproti asp.....	32
4.2 PHP.....	33
4.2.1 Směrování PHP .....	33
4.2.2 Rozšíření objektového modelu.....	34
4.2.3 Podpora různých kódování řetězců .....	36
4.2.4 Dodržování zásad při pojmenovávání.....	36
4.2.5 Chaotická implementace výjimek .....	37
4.2.6 Ztracená příležitost.....	37
<b>II PRAKTICKÁ ČÁST .....</b>	<b>38</b>
<b>5 PŘÍKLADY WEBOVÝCH PROHLÍŽEČŮ.....</b>	<b>39</b>
5.1 GRAFICKÉ WEBOVÉ PROHLÍŽEČE .....	39
5.2 TEXTOVÉ WEBOVÉ PROHLÍŽEČE .....	40
5.3 PŘEHLED VERZÍ NEJPOUŽÍVANĚJŠÍCH PROHLÍŽEČŮ .....	40
<b>6 SOUČASNÉ WEBOVÉ TECHNOLOGIE.....</b>	<b>43</b>
6.1 WEBOVÉ PROHLÍŽEČE .....	43
6.1.1 Opera .....	43

6.1.2	Mozilla .....	43
6.1.3	Internet Explorer .....	44
6.1.4	Netscape Navigator .....	44
6.1.5	Kritika .....	44
6.2	CO HÝBE DNEŠNÍMI WEBOVÝMI STRÁNKAMI .....	45
6.2.1	Java & JavaScript.....	46
6.2.2	Kreativita jménem Flash .....	46
6.2.3	Interaktivita sem, interaktivita tam... ..	47
<b>7</b>	<b>BEZPEČNOST WEBU .....</b>	<b>48</b>
7.1	JAVASCRIPT.....	48
7.1.1	Bezpečnostní rizika Javascriptu .....	48
7.1.2	Cross-frame scripting .....	50
7.2	COOKIES.....	51
7.2.1	Bezpečnost cookies a chyby webových prohlížečů .....	51
7.2.2	Ztráta anonymity měřením návštěvnosti webových stránek.....	51
7.3	SESSION.....	52
7.3.1	Prozrazení sessionID .....	52
7.3.2	Metody předávání sessionID.....	53
7.3.3	Zcizení session .....	54
7.4	SQL DATABÁZE .....	54
7.4.1	Vkládání cizích SQL podmínek (SQL injection).....	54
<b>8</b>	<b>VÝHODY WEBOVÝCH TECHNOLOGIÍ.....</b>	<b>56</b>
8.1	SOUHRNNĚ: .....	56
8.1.1	Weby sociálních sítí .....	56
8.1.2	Blogy .....	57
8.1.3	Podcasting .....	57
8.1.4	Inteligentní tiskové zprávy .....	57
8.1.5	Cílená propagace.....	58
<b>9</b>	<b>NEVÝHODY WEBOVÝCH TECHNOLOGIÍ .....</b>	<b>59</b>
9.1	SOUHRNNĚ: .....	59
9.1.1	Statické webové stránky.....	59
9.1.2	Dynamické www stránky .....	59
<b>10</b>	<b>NOVÉ TRENDY VE VÝVOJI WEBOVÝCH PROHLÍŽEČŮ.....</b>	<b>61</b>
10.1	WEB 2.0 .....	62
10.2	WEBOVÉ INŽENÝRSTVÍ.....	62
10.3	VÝVOJ SOFTWAREOVÝCH VS. WEBOVÝCH SYSTÉMŮ .....	64
<b>11</b>	<b>NOVÉ TRENDY VE VÝVOJI WEBDESIGNU .....</b>	<b>66</b>

---

11.1	TŘI PILÍŘE MODERNÍHO WEBU .....	67
11.2	DŮRAZ NA UŽIVATELSKÝ PROŽITEK .....	67
11.3	DŮRAZ NA PŘÍSTUPNOST .....	68
11.4	DŮRAZ NA EKONOMICKOU NÁVRATNOST A EFEKTIVITU .....	68
11.5	BUDOUCNOST WEBOVÝCH TECHNOLOGIÍ .....	69
<b>ZÁVĚR.....</b>		<b>71</b>
<b>ZÁVĚR V ANGLIČTINĚ.....</b>		<b>73</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>		<b>75</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>		<b>76</b>
<b>SEZNAM OBRÁZKŮ .....</b>		<b>79</b>
<b>SEZNAM TABULEK.....</b>		<b>80</b>

## ÚVOD

Data jsou na webu až dosud prezentována s předpokladem, že je jejich uživatelem člověk. Datové formáty zdrojů na webu, jako je jazyk HTML nebo PDF, popisují pouze formální prezentační stránku nalezené informace, nikoliv tu významovou. Význam dat pro lidského uživatele vyplývá i z jejich obsahu, totéž však neplatí pro počítače. Počítače významu informace na webu nerozumějí. Podstatnou skutečností se stává fakt, že uživatelem webu už není jenom člověk, ale i počítače.

Tradiční vývoj webových uživatelských rozhraní podnikových aplikací vyžaduje rozsáhlé znalosti vývojářů v oblasti programování webové prezentační vrstvy, použité technologie webového portálu a vlastních podnikových aplikací. Například společnost SAP nabízí novou technologii pro efektivní vývoj webových aplikací skládáním bez jediné řádky programového kódu.

Uživatelé informačních systémů v podnicích a organizacích stále více preferují intuitivní uživatelská rozhraní webových aplikací integrovaných do webového portálu. Vyšší pohodlí a produktivitu práce koncových uživatelů paradoxně vyvažuje výrazně pracnější vývoj. Tradiční vývoj webových uživatelských rozhraní nad stávajícími podnikovými aplikacemi je složitý proces, který zahrnuje definici procesů, datových objektů, kódování ve zvoleném programovacím jazyce a vyčerpávající ladění. Zároveň vyžaduje detailní znalost podnikových aplikací a použité portálové technologie. IT manažeři tak stojí před nelehkým úkolem nalezení vývojářů s velmi rozsáhlým portfoliem znalostí.

Mnozí říkají, že web vstoupil do další etapy svého vývoje. Nová generace služeb online založená na sdílení a spolupráci v širokém měřítku bývá označována jako Web 2.0. Jsou to systémy, které jsou tím inteligentnější, čím více lidí je používá.

Vývojáři sice s jistotou neví, jaká bude další významná aplikace pro web, jsou však přesvědčeni, že půjde o mobilní web. Vývoj jednou dospěje do bodu, kdy mobilní zařízení budou tvořit základ pro přístup k webu.

Aby všechny ty věci mohly správně fungovat, je naprosto nezbytné, aby jedna strana mohla bezpečně zjistit veřejný klíč druhé strany. Proto se mimo jiné zabývají způsoby, jak zajistit, aby to nebylo nákladné. Kdyby umožnili, aby zde jedna firma získala monopol, přišlo by nás to velmi draho. Společnosti se dnes snaží zajistit, aby připojení



bylo dostupné pro všechny, bez ohledu na to, kde se nacházejí a jakou technologii používají.

Zajímavý je zejména poslední trend v metodice vývoje, a to je zpřístupnění stejných vizuálních vývojových prostředků pro vývoj webových aplikací, jaké používají již několik let vývojáři desktopových aplikací. Výsledkem by mělo být stejné vývojové prostředí a stejné možnosti v programování - tedy pokročilejší techniky OOP včetně dědičnosti. Hlavními hráči zde jsou SUN s javovými technologiemi a Microsoft s chystanou technologií .NET. Bez zajímavosti není ani to, že obě tyto technologie se již nespolehnají pouze na interpretovaný kód, ale používají také předkompilovaný mezikód.

## I. TEORETICKÁ ČÁST

# 1 HISTORIE INTERNETU

## 1.1 USA

Zhruba před třiceti lety stála společnost RAND (přední americké mozkové centrum z období studené války) před zvláštním strategickým problémem: jak zajistit úspěšnou komunikaci úřadů USA po nukleární válce?

V průběhu 60-70-tých let byla tato spletitá koncepce decentralizované odolné paketově - orientované sítě rozpracována organizací RAND spolu s univerzitami MIT a Los Angeles (UCLA). Krátce poté se Pentagonská agentura ARPA (Advanced Research Projects Agency) rozhodla financovat obdobný, avšak větší a ambicióznější projekt v USA. Uzly této sítě se měly stát vysoce výkonné superpočítače (měřeno tehdejšími měřítky).

Na podzim 1969 byl instalován první uzel sítě v UCLA a koncem téhož roku byly na světě čtyři uzly, které vytvořily zárodek sítě, pojmenované podle svého sponzora ARPANET. Tyto čtyři počítače mohly vzájemně přenášet data přes dedikované rychlé přenosové linky; mohly být dokonce i vzdáleně programovány z jiných uzlů. V roce 1971 měl již ARPANET patnáct uzlů, o rok později třicet sedm. A byla to dobrá síť!

Během 70-tých let se síť ARPANET stále více rozrůstala; její decentralizovaná struktura tuto expanzi jen usnadňovala. Na rozdíl od standardních firemních počítačových sítí mohl ARPANET připojovat mnoho různých typů počítačů. Stačilo, aby tyto počítače rozuměly paketově - orientovanému protokolu nové "anarchistické" sítě, vše ostatní bylo irelevantní.

Již od roku 1977 začal být TCP/IP používán jinými sítěmi pro připojování k ARPANETu. Samotný ARPANET zůstával pod pevným řízením přinejmenším do roku 1983, kdy se jeho vojenský segment oddělil a vytvořil samostatnou vojenskou síť MILNET.

Jak ubíhala 70. a 80. léta, přístup k výkonným počítačům získávaly mnohé velmi rozdílné sociální skupiny lidí. S tím, jak se používání TCP/IP stávalo čím dál běžnějším, se řada kompletních sítí dostávala do digitálního objektu Internetu. Protože software TCP/IP byl volně dostupný (public-domain) a základní technologie byla decentralizovaná a ze své podstaty přímo poněkud anarchistická, bylo obtížné zastavit lidi v invazi na síť.

Ve skutečnosti nikdo neměl zájem bránit jim v připojování na rozšiřující se komplex sítí, který začal být všeobecně znám pod označením Internet.

Připojování k Internetu stálo daňové poplatníky jen málo nebo dokonce vůbec nic, protože každý uzel byl nezávislý a musel si sám zajistit své vlastní financování a potřebné technické vybavení. Čím více, tím lépe; podobně jako telefonní síť, i počítačová síť se postupně stávala tím užitečnější, čím větší a větší oblasti lidí a zdrojů zahrnovala. Fax je cenný tehdy, jestliže všichni jej mají také; dokud tomu tak není, je pouhou kuriozitou. Také ARPANET byl chvíli kuriozitou; pak se však síťování počítačů stalo naprostou nezbytností.

V roce 1984 vstoupila do hry Národní vědecká nadace USA (NSF - National Science Foundation) prostřednictvím svého úřadu pro pokročilé vědecké výpočty, který inicioval vývoj sítě NSFNET. Síť NSFNET dnes poskytuje hlavní páteř pro komunikační služby Internetu ve Spojených státech; při přenosových rychlostech stovek megabitů za sekundu přenáší každý měsíc desítky miliard paketů mezi sítěmi, které propojuje.

Uzly v Internetu byly rozčleněny do skupin; mimo-americké počítače a několik amerických uzlů zvolily označování podle svého geografického umístění, většina amerických počítačů byla však seskupena do šesti historicky základních internetovských domén: gov, mil, edu, com, org a net; domény gov, mil a edu označují vládní (governmental), vojenské (military) a vzdělávací (educational) instituce, jež byly přirozeně průkopníky od samého začátku, kdy ARPANET vznikla jako high-tech výzkumné cvičení v oblasti národní bezpečnosti Spojených států.

Samotná síť ARPANET formálně zanikla v roce 1983 jako šťastná oběť svého zdrcujícího úspěchu. Její uživatelé to prakticky ani nezaznamenali, neboť ARPANETovské služby nejenže fungovaly dál, ale navíc byly a jsou neustále zdokonalovány. Použití standardu TCP/IP pro počítačové síťování se stalo globální záležitostí. Dnes má Internet 4 miliony uzlů rozprostřených ve 40.000 sítích prakticky všech zemí světa a každý den se připojují další a další. Desítky milionů lidí používají denně tuto "matku všech počítačových sítí".

## 1.2 Evropa

Není tomu nijak dlouho, co internet mohl poprvé okusit český člověk, teprve něco málo přes deset let. Český internet poznal rychlý rozvoj a dohnal, ne-li předešel, úroveň světového internetu, i když je pravda, že český internet se rozvíjí víceméně bez výraznější podpory státních orgánů (neberu v potaz zavádění internetu do škol), zatímco v USA je internet chráněn jako národní bohatství.

## 1.3 Česká republika

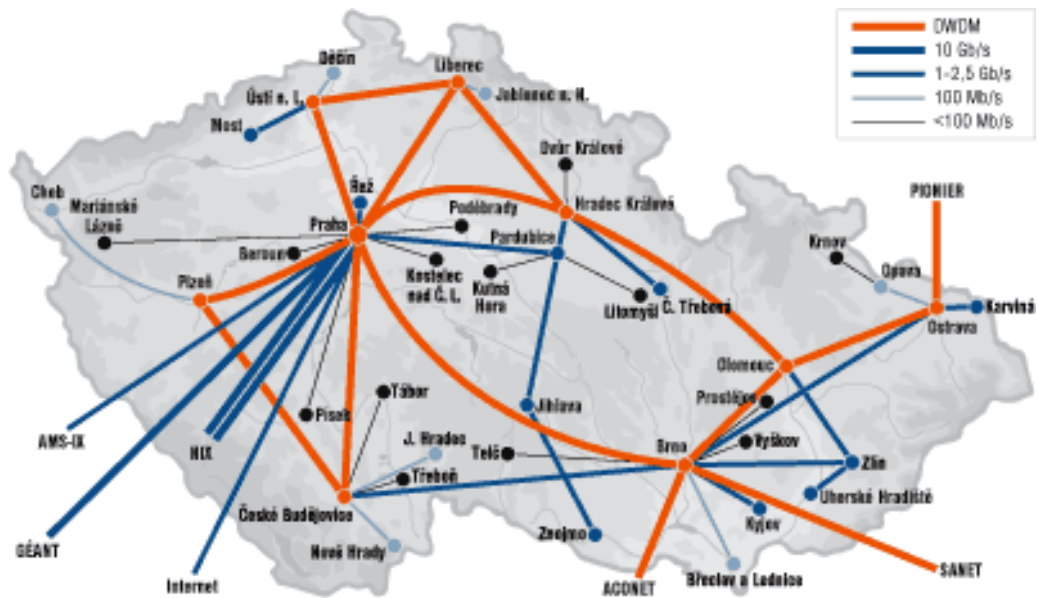
Historie českého Internetu se začíná psát počátkem roku 1990. V té době ještě v bývalém Československu neexistovala žádná pevná linka vyjma telefonních, a tak se první pokusy o vytvoření počítačové sítě děly pomocí komutovaných linek veřejné telefonní sítě. V březnu toho roku se do naší republiky dostává síť FIDO a následně v květnu pak síť EUnet, která propojovala zejména Unixové stanice. Ale už v říjnu roku 1990 k nám přichází síť EARN (European Academic and Research Network). Prvním uzlem této sítě byl střediskový počítač IBM 4381, který byl umístěn v Oblastním Výpočetním Centru (OVC) ČVUT v Praze. Přenosová rychlost linky směřující z pražského uzlu do rakouského národního uzlu EARN v Linci byla 9600 bps.

První pokusy o připojení do Internetu proběhly v listopadových dnech roku 1991. Linka vedla z Prahy (ČVUT) do internetového uzlu v Linci. Zprvu šlo pouze o komutovaný spoj, jenž byl později nahrazen pevnou linkou, která využívala již dříve zavedeného spojení sítě EARN. Do této doby se také datuje jakési neoficiální připojení Československa k Internetu.

Dne 13.2.1992 probíhá slavnostní oficiální připojení naší republiky k Internetu. K tomuto aktu nedochází nikde jinde, než na pražském ČVUT.

Již v roce 1991 byl podáván návrh na vybudování celorepublikové páteřní sítě. Ta měla propojovat všechna tuzemská akademická centra a dále by pak měly Internet rozvádět metropolitní sítě. Na základě tohoto návrhu vnikly v Československu dva projekty na vybudování národních páteřních sítí, slovenské a české, přičemž připojení těchto dvou páteří bylo součástí projektu české strany (konkrétně šlo o spoj vedený z Prahy do Bratislavy), Český projekt dostal jméno FESNET (Federal Educational and Scientific Network). V červnu 1992 české ministerstvo školství schválilo projekt a uvolnilo na něj

20 milionů korun. V průběhu roku 1992 se písmeno F v názvu FESNET změnilo na C a tím vlastně vznikl CESNET (Czech Educational and Scientific NETwork). Na Slovensku se souběžně začala budovat síť SANET (Slovak Academic NETwork).



Obr. 1. Mapa sítě CESNET v České republice

Síť CESNET byla zpočátku realizována hvězdicovou topologií se středovými uzly v Brně a v Praze. Ty byly propojeny pevnou linkou o rychlosti 64 kbps. K těmto dvěma uzlům byly připojovány další uzly umístěné v našich akademických městech. Postupně byly připojeny České Budějovice, Hradec Králové, Liberec, Plzeň a další. V březnu roku 1993 byly uzly CESNETu rozmístěny už v 11 městech.



Obr. 2. Mapa sítě CESNET v zahraničí

Na přelomu let 1994 a 1995 byla komunikační infrastruktura CESNETu v podstatě dobudována a pozornost se přesunula především na zvyšování přenosových rychlostí a s ním související zlepšování spolehlivosti. Pro představu uvádím přenosové rychlosti některých spojů CESNETu v lednu roku 1995.

**512kbps :**

Praha -> Amsterdam

**128kbps :**

Praha -> Vídeň

Praha -> Brno

Praha -> Liberec

Liberec -> Ostrava

Brno -> Olomouc

Olomouc -> Ostrava

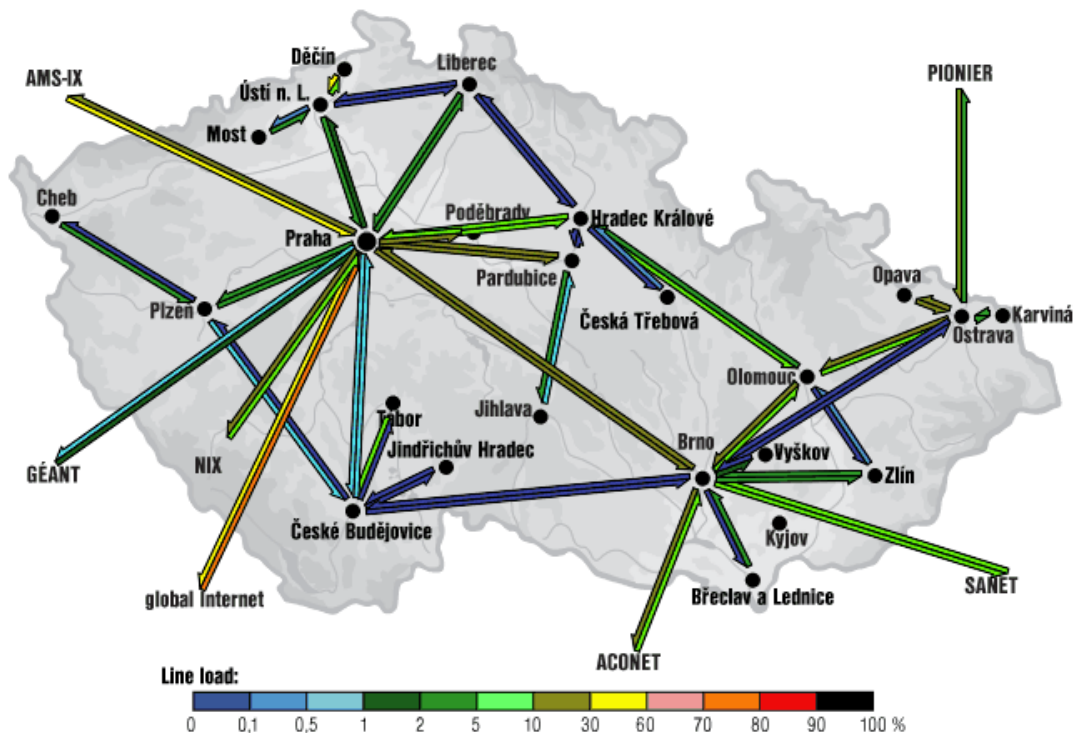
64kbps :

Praha -> Plzeň

Praha -> České Budějovice

Plzeň -> České Budějovice

Většina ostatních linek fungovala na 19,2 kbps



Obr. 3. Aktuální zatížení páteřních linek CESNETu

První čistě komerční firmou, která začala zprostředkovávat připojení k Internetu, byla firma COnet, která začala provozovat síť CZnet, jež vznikla z pražského uzlu sítě EUnet. Tato síť disponovala 64kbitovou linkou vedoucí z Prahy do Amsterdamu.

Zajímavé by možná bylo podívat se, v jakých cenových relacích se v tehdejší době poskytování připojení k Internetu pohybovalo.

#### 1.4 Projekt TEN - 34 CZ

Jak již bylo zmíněno, přibližně od roku 1995 s nástupem komercializace Internetu, zažívá toto médium mohutný rozvoj. Tento rozvoj se ovšem netýká pouze komerční oblasti. Už v roce 1994 se Česká republika zapojuje do projektu TEN-34 (Trans-European



Network Interconnect at 34 Mbit/s). Jak již název napovídá, cílem projektu bylo vybudovat vysokorychlostní celoevropskou síť, která by propojovala národní vědecké a univerzitní sítě rychlostí až 34 Mbit/s. V České republice byl projekt TEN-34 CZ schválen ministerstvem školství, mládeže a tělovýchovy dne 3. října 1996. Řešitelem projektu se stal CESNET, z. s. p. o., tedy sdružení zahrnující všechny vysoké školy a Akademii věd ČR. Koncem roku 1996 byl uveden do provozu první radioreleový okruh o rychlosti 34Mb/s z Prahy do Brna, jenž byl dodán českými radiokomunikacemi.

V únoru 1997 byla uvedena do provozu dvojice mezinárodních datových okruhů dodaných společnostmi SPT Telecom. Jednalo se o linky Praha - Mnichov a Praha - Stockholm, každá o kapacitě 2 Mb/s. Začátkem srpna byl zahájen provoz dalšího mezinárodního datového okruhu o kapacitě 34 Mb/s Praha - Frankfurt, jež byl dodán firmou Global One Communications. Díky zprovoznění tohoto okruhu se stal CESNET oficiálním členem celoevropského projektu TEN-34.

V polovině roku 1998 byla k TEN-34 CZ připojena města Ústí nad Labem (k pražskému uzlu), Zlín (k brněnskému uzlu) Cheb (k uzlu v Plzni) a Karviná s Opavou (k uzlu v Ostravě). Tím dosáhl počet uzlů připojených k TEN-34 CZ počtu 14. V červenci roku 1998 byla zvýšena přenosová rychlost páteřní linky Praha - Brno z 34Mb/s na 155Mb/s. Také byla zvýšena kapacita zahraničních spojů na požadovaných 34Mb/s. Konkrétně se jednalo o 16Mb/s linku z Prahy do Frankfurtu a 18Mb/s linku z Prahy do New Yorku. Dále byly v roce 1998 připojeny k TEN-34 CZ (kromě většiny vysokých škol a Akademie věd) také významnější knihovny a fakultní nemocnice. Celoevropský projekt TEN-34 končí dne 30.11.1998 a na něj navazuje projekt QUANTUM, který si klade za cíl vybudovat evropskou akademickou síť o přenosové kapacitě 155Mb/s. Sdružení CESNET se stalo členem projektu QUANTUM již v červenci roku 1998.

Shrňme-li tedy výsledky, kterých projekt TEN-34 CZ dosáhl, vidíme, že během necelých tří let se zvýšila kapacita mezinárodních linek z původních 2Mb/s na 34Mb/s a kapacita nejdůležitějšího vnitrostátního spoje z 3Mb/s na 155Mb/s. Kromě toho se oddělily původně sdílené linky komerční a akademické sítě.

## 1.5 Projekt QUANTUM (TEN-155)

Jak již bylo řečeno, projekt QUAMTUM se stal následníkem TEN-34. V rámci projektu se dále pracuje na zvýšení spolehlivosti a bezproblémovém chodu sítě TEN-155.

V roce 1999 dochází k vylepšení spojení do zahraničí, kdy jsou uvedeny v chod 2 nezávislé linky, jedna s napojením na evropskou TEN-155 a druhá s připojením do USA přes EBONE. Celková kapacita zahraničních linek tak vzrostla z 34Mb/s na 79Mb/s.

Dne 10. února 2000 byl zahájen testovací provoz linky s rychlostí 2,5 Gb/s na trase Praha - Brno.

## 1.6 Ostatní sítě na území České republiky

Kromě CESNETu a jím provozované akademické sítě TEN-155 se na české scéně postupně objevilo množství dalších sítí a poskytovatelů přístupu k Internetu. Nejdůležitějšími z nich jsou primární poskytovatelé, tedy ti, kteří mají vlastní nebo pronajatou linku do zahraničí. Nejvýznamnější poskytovatelé a jejich linky do zahraničí, uvádím v následujícím přehledu. Tyto informace byly aktuální k datu 31.1.2007.

*Tab. 1. Přehled nejvýznamnějších poskytovatelů a jejich linky*

Poskytovatelé	Linky
AT&T Unisource	- Praha - Amsterdam
komerční část CESNETu prodána Contactelu	- 16Mb/s na pražský uzel Ebone - satelitní linka do USA
Contactel	- 3x2Mb/s na TeleDenmark - 2Mb/s na pražský uzel Ebone
CZCOM	- 384kb/s do Nitedalu (Norsko) - 1Mb/s na pražský uzel Ebone
CZech On Line (VOL)	- 10Mb/s ATM na síť Teleglobe do New Yorku - 2Mb/s na pražský uzel Ebone
DirectNet	- 2Mb/s do New Yorku

Poskytovatelé	Linky
Ebone	- 155Mb/s do sítě Ebone na uzel v Mnichově

Global One	- 34Mb/s ATM do sítě Deutsche Telecom - 4Mbit/s do páteře GlobalOne (Pennsauken a Londýn)
GTS INEC	- 1Mb/s satelitní linka na EUNET (Washington) - 4Mb/s na pražský uzel Ebone - 2Mb/s na pražský uzel Global One
IBM	- 4Mb/s do Ehnigenu - 2Mb/s do Portmouthu - 768Kb/s do Bratislavy
Internet CZ (EUnet)	- 34Mb/s satelitní linka Praha – Amsterdam - 34Mb/s pozemní linka Praha – Amsterdam - 64Kb/s do Bratislavy
InWay	- 2Mb/s do sítě EUNETu na uzel ve Frankfurtu
LUKO CZECH-NET	- 4Mb/s do sítě AUCS na uzel ve Frankfurtu
PVT	- 2Mb/s do sítě EUNET do USA - 8Mb/s na pražského uzlu Global One
Telecom	- 8Mb/s do sítě Teleglobe na uzel v New Yorku - 6Mb/s na pražský uzel Global One
Telenor	- 512Kb/s satelitní linka do Nittedalu (Norsko)
Telenor Internet	- 34Mb/s do sítě NextBone na uzel ve Frankfurtu - 34Mb/s na Slovensko - 3,5Mb/s na pražský uzel Ebone
CZCOM	- 384kb/s do Nitedalu (Norsko) - 1Mb/s na pražský uzel Ebone
CZech On Line (VOL)	- 10Mb/s ATM na síť Teleglobe do New Yorku - 2Mb/s na pražský uzel Ebone

K těmto sítím s mezinárodní konektivitou je samozřejmě připojeno velké množství tuzemských sítí a sekundárních poskytovatelů. K 31.1.2000 bylo v naší republice více než 350 poskytovatelů přístupu k Internetu (ISP - Internet Service Provider).

## 1.7 Peering - spása domácího Internetu

Důležitým aspektem tuzemského Internetu je otázka takzvaného peeringu. O co se jedná? Jak již bylo výše zmíněno, v naší republice vstoupilo na trh množství primárních poskytovatelů připojení k Internetu, tedy těch, kteří mají vlastní linku do zahraničí. Problém nastává ve chvíli, kdy spolu chtějí komunikovat dva tuzemští uživatelé připojení k různým poskytovatelům. Kudy budou probíhat jimi vyměňovaná data? Pokud tito dva poskytovatelé nejsou vzájemně propojení (nepeerují, nemají spolu peering), jdou data nebohých uživatelů nejprve linkou jednoho poskytovatele ven za hranice a linkou druhého zase zpět. Jaké nevýhody tento přístup přináší, je na první pohled zřejmé.

Nejenže data těchto dvou účastníků, fyzicky od sebe vzdálených třeba jen pár kilometrů, putují mnohdy přes půlku světa a tím se do značné míry zpožďují, ale jsou především zahlcovány linky ven, jejichž průchodnost bývá vždycky klíčovou a to nemluvím o tom, že provoz zahraniční linky bývá pro providery dražší záležitostí, než provoz linky domácí. Proto se už od chvíle, kdy se na scéně objevuje více providerů, začíná mluvit o naléhavé potřebě peeringu.

Velcí provideři, kteří byli na trhu první (CESNET a Eunet (dříve Internet CZ, předtím COnet)), měli mezi sebou peering už od velmi brzkých dob a to díky fyzické blízkosti jejich sítí přímo na úrovni LAN (Ethernetu). První globálnější pokus jak řešit svízelnou otázku peeringu byl učiněn v létě roku 1996. Konkrétně 30. srpna toho roku bylo založeno zájmové sdružení právnických osob s názvem NIX (Neutral Internet eXchange). Zakládajícími členy byli CESNET, Czech On Line, Datac, GTS CzechCom, IBM, Internet CZ a PVT. Telecom, který také pracoval na přípravách, nakonec zakládající smlouvu nepodepsal. Každý ze členů měl právo na dvě datové linky o maximální kapacitě 2Mb/s. To znamená o celkové kapacitě max. 4Mb/s. Tyto spoje byly zakončeny směrovači a ústily do společného segmentu 10Mbitového Ethernetu. Pět členů se rozhodlo připojit mikrovlnným spojem (Czech On Line, Datac, GTC, IBM a PVT) a dva zbylí (CESNET a Internet CZ) se rozhodli pro spoj pevnou linkou. Jako první si linku přivedl Datac

(10.1.1997). Dne 4.2.1997 přibyly další dva mikrovlnné přípoje (PVT, GTS) a jelikož tyto tři poskytovatelé měli dohodu o vzájemném peeringu již sepsanou, téhož dne začal fungovat peering na bázi NIXu. 7.2.1997 byla přivedena linka IBM atd.

Postupem času se do sdružení NIX.CZ připojovali další členové:

LUKO CZECH-NET (1.4.1997)

Telenor Internet (1.4.1997)

NetForce (1.6.1997)

MOPOS (1.11.1997)

TELENOR ČR (1.11.1997)

CZCOM (1.11.1997)

Global One Communications (1.5.1998)

InWay (1.5.1998)

ČESKÝ TELECOM (1.5.1998)

INEC (1.7.1998)

IPEX (1.4.1999)

DirectNet-EOS (1.4.1999)

Contactel (1.7.1999)

PragoNet (1.1.2000)

## 2 HISTORIE WEBOVÝCH STANDARDŮ

Theodor Nelson začal pracovat na realizaci nelineárního spojování dokumentů na počítačích už v 60. letech 20. století. Byl to on, kdo první přišel s pojmem hypertext. Odtud již nebylo daleko ke zkratce HTML, tj. HyperText Markup Language = „hypertextový značkovací jazyk“

Roku 1986 vzniká SGML (Standard Generalized Markup Language), definovaný v normě ISO 8879. Tento obecný značkovací jazyk umožňuje definici vlastních (podřízených) značkovacích jazyků na základě vlastní definice typu dokumentu - DTD. Jazyk HTML je jednou z aplikací SGML, každá verze HTML používá značky tak, jak je popsáno v každé příslušné DTD.

Následuje vývoj HTML, jak jej v publikaci Dynamické HTML popisuje Pavel Mikle, (Unis Publishing 1997).

1989

- Tim Berners-Lee přichází s projektem vytvoření distribuovaného hypertextového systému
- ve výzkumných laboratořích CERN ve Švýcarsku je zahájen projekt WWW

1991, 1992

- zveřejněna neformální specifikace HTML
- první veřejně dostupná verze prohlížeče; pouze textový režim

1993

- na světě pracuje kolem padesáti serverů
- dokončen první grafický prohlížeč dokumentů NCSA Mosaic; pro prostředí X-Window
- návrh jazyka HTML verze 2.0

1994

- první mezinárodní konference věnovaná systému WWW

- autor programu Mosaic zakládá společnost Mosaic Communications Corp.; tato uvede za krátkou dobu prohlížeč pod názvem Netscape
- v září založena organizace World Wide Web Consortium (dále jen W3C)
- CERN předává vývoj WWW francouzskému institutu INRIA

1995

- WWW se rozšiřuje
- oficiální specifikace HTML 2.0 (INRIA)
- Netscape uvádí neoficiální rozšíření HTML, známé jako HTML 3.0

1996

- oficiální specifikace HTML 3.2 (W3C); v porovnání s HTML 3.0 je chudší
- Microsoft vypouští svůj první prohlížeč zdarma, Internet Explorer 3.0; první podpora CSS
- specifikace CSS level 1

1997

- specifikace HTML 4.0 (W3C); přidány rámy a plovoucí rámy, vylepšeny formuláře a tabulky, větší podpora skriptů, nové řádkové elementy (mimo jiné i ABBR, doposud nepodporovaný Internet Explorerem)
- Microsoft vydává Internet Explorer 4.0; směr - DHTML

1998

- Microsoft vydává Internet Explorer 5.0
- první specifikace XML 1.0
- specifikace CSS level 2

1999

- specifikace HTML 4.01

2000

- první specifikace XHTML 1.0
- specifikace XML 1.0 (Second edition)

2001

- Microsoft vydává Internet Explorer 6
- specifikace XHTML 1.1

2002

- reformulace XHTML 1.0

2003

- pracovní návrh XHTML 2.0
- pracovní návrh revize CSS level 2.1
- pracovní návrh CSS level 3

Webové standardy jsou doporučení a normy pro tvorbu webových stránek. Jejich dodržení zaručí vysoký úspěch webové prezentace.

Webové standardy jsou tedy specifikace a pravidla pro tvorbu webových stránek. Mají za cíl rozvíjet sílu Internetu jako informačního média, zajistit všeobecně přístupné, přehledné a bezbariérové Internetové dokumenty. Webové standardy navrhuje organizace The World Wide Web Consortium (W3C).

K definici struktury webových stránek se používá jazyk XHTML nebo starší HTML. Pomocí speciálních značek v dokumentu sdělíme webovému prohlížeči, kde se nachází jaký prvek (nadpisy, obrázky, tučné písmo, odkazy). Tím je připravena struktura dokumentu.

Pro vizuální formátování se používají kaskádové styly, pomocí kterých je zajištěno oddělení obsahu od formy. Výsledkem je datově menší a přehledný zdrojový kód. Takové webové stránky se rychleji načítají a je možné jednoduše upravovat jejich vzhled.

## 2.1 Proč dodržovat webové standardy

Stránky, které dodržují webové standardy, jsou kompatibilní se všemi prohlížeči - staršími i moderními. Proto se pak nemůže stát, že webová prezentace vypadá na každém počítači a v každém prohlížeči trochu jinak.



- je zajištěna zpětná i dopředná kompatibilita, v moderních prohlížečích a přístrojích se zobrazí plně grafické stránky, starší prohlížeče zobrazí vizuálně neformátovanou, ale stále strukturovanou stránku
- webové stránky, které dodržují webové standardy, se rychleji načítají
- vyhledávače si stránky dodržující standardy lépe zaindexují a lépe umístí ve výsledcích vyhledávání
- jednoduché úpravy vzhledu stránek

Na internetu je dnes zveřejněno téměř vše. Problém je, kde to najít. S obrovským rozvojem internetu přišel taky obrovský chaos, který je částečně způsoben právě nedodržováním a odklonem od původní myšlenky webových standardů. Webové standardy mají za cíl plně rozvíjet potenciál internetu a pomocí sémantiky vytvořit síť, ve které podle určitých pravidel snadněji najdete to, co hledáte.

HTML se už ve své první verzi, díky své jednoduchosti, stal velmi populární. Původně byl HTML určen pro zobrazování struktury dokumentu bez vážnějšího důrazu na vizuální a grafickou interpretaci.

Při vývoji HTML však došlo, na nátlak ze strany tvůrců stránek a prohlížečů, k jistému odklonu tím, že se do nových verzí HTML začaly postupně přidávat různé formátovací značky pro vzhled a interaktivitu.

Klíčová chyba nastala, když se jako nástroj pro design webu začaly používat tabulky. Tabulka měla a má sloužit pro prezentaci tabulkových dat. Nikoliv pro utváření vzhledu stránky. To způsobilo, že se z jednoduchého značkovacího jazyka stal nástroj na tvorbu prezentací a formátovací značky pak tvořily podstatný díl obsahu webových stránek (specifikace HTML 3.2 z roku 1996). Tím se vytratila struktura a čitelnost, která byla hlavním záměrem původního HTML a internet do jisté míry ovládl chaos.

Na přelomu století se ukazuje, že HTML je díky svému odklonu od původní myšlenky příliš rozvolněný formát a potřeba více striktního formátu začíná být nutností. V roce 2000 tak přichází na svět první specifikace XHTML.

## 2.2 Výhody webových standardů

Hlavními výhodami moderních webových standardů (XHTML) je to, že došlo k oddělení struktury a obsahu webových dokumentů od jejich vzhledu. To způsobilo, že dokumenty dodržující tyto standardy jsou co se týče objemu dat menší (a tudíž se rychleji načítají) a přehlednější.

Oddělením obsahu (textu) od vzhledu (soubory CSS) je textová část webové stránky rovněž použitelná pro uživatele s malými obrazovkami PDA, organizérů a mobilních telefonů, pro software screenreaderů zrakově postižených lidí a v neposlední řadě pro vyhledávače.

## 2.3 Dodržování webových standardů v praxi

I přes výhody, které webové standardy přináší, není jejich dodržování samozřejmostí. Svou vinu na tom mají i internetové prohlížeče, které zatím stále fungují tak, že dovolují tvůrcům stránek dělat v kódu chyby. Mnoho tvůrců webu se pak spokojí s tím, když jim stránky fungují alespoň v Internet Exploreru (s argumentem, že je to stejně ten nejpoužívanější prohlížeč). Tím se ale stávají nepřístupné pro jiná zařízení a zejména pro zmíněné internetové vyhledávače.

## 3 HISTORIE A VÝVOJ WEBOVÝCH PROHLÍŽEČŮ

### 3.1 Co to vlastně je?

Webový prohlížeč je program, který umožňuje uživateli zobrazit a upravovat HTML kód, který je uložen v jakémkoliv dokumentu na webovém serveru. Největší, pomocí sítě propojená kolekce těchto dokumentů, se nazývá World Wide Web (zkráceně WWW). Komunikace mezi webovým serverem a prohlížečem využívá standardně protokol pro přenos textu HTTP. Většina prohlížečů také podporuje další protokoly, jako jsou FTP, Gopher nebo HTTPS (tj. šifrovaná verze HTTP). Webové prohlížeče jsou schopné zobrazovat soubory uložené také v jiných formátech než je HTML kód. Toto umožňuje zařazovat přímo k webové stránce obrázky, animace, video, zvukové soubory nebo je pomocí webové stránky zpřístupnit. Některé z novějších prohlížečů mají v sobě zabudovány komponenty pro podporu Usenet zpráv, e-mailu přes NNTP, IMAP a POP protokoly. Většina webových prohlížečů umožňuje ukládat záložky na oblíbené, či navštívené stránky.

### 3.2 Historie

{1989 -1992}

V roce 1989 ve švýcarském výzkumném středisku CERN v Ženevě, které bylo jako jedno z mála připojeno k tehdy se rodícímu Internetu, definoval Tim Berners-Lee první hypertextový systém pro CERN. Dnes bychom řekli, že se jednalo o první intranet na světě.

O rok později napsal první program pro tvorbu primitivních hypertextových stránek a pro systém, který tyto stránky zobrazoval, zvolil název World-Wide Web. Jeho nápad umožňoval vědcům ze vzdálených míst zeměkoule organizovat a uspořádat informace, ovšem na rozdíl od jednoduchého zpřístupnění velkého množství dokumentů o výzkumech v podobě souborů, které mohou být staženy do jednotlivých počítačů, navrhnul, že můžeme zařídit, aby texty v souborech na sebe přímo odkazovaly. Jinými slovy, aby bylo možné vytvořit křížové odkazy z jednoho dokumentu na jiný. To by znamenalo, že během čtení jednoho dokumentu můžeme rychle zobrazit část dokumentu souvisejícího, který obsahuje přímo příslušný text nebo diagram.

{1992 -1993}

Na druhé straně světa, ve výzkumném ústavu na univerzitě Illinois (NCSA), se jeho práce zalíbila natolik, že se na základě 2 dostupných verzí webových prohlížečů z laboratoře CERN rozhodli vytvořit svůj vlastní prohlížeč, který nazvali Mosaic. Byl určen pro všechny běžné platformy (Unix, PC/Windows, Macintosh) a jeho první alfa verze spatřila světlo světa v únoru 1993.

Zhruba ve stejné době se objevuje také prohlížeč Alpha od firmy Macintosh, řádkový prohlížeč pracující přes Telnet. V březnu 1993 je vydán díky Lou Montulliovi textový prohlížeč Lynx zobrazující pouze HTML text je určený pro terminály a pro počítače, které používaly systém DOS. Do této doby také můžeme zařadit prohlížeče Viola a Midas uveřejněné v lednu 1993 pro systémy Unix. Během roku 1993 a začátkem roku 1994 si mnoho prohlížečů začalo přidávat do HTML své vlastní prvky, čímž se tento jazyk stal špatně definovaným. V úsilí vytvořit pořádek ve vzniklém chaosu, byly všechny značky jazyka HTML uspořádány do jednoho pracovního dokumentu, který se specifikoval jako standard HTML2.

{1994}

Jeden z hlavních tvůrců Mosaicu Marc Andressen opouští po neshodách NCSA a společně s Jimem Clarkem zakládají společnost Mosaic Communications (později přejmenováno na Netscape Communications), která v prosinci 1994 vydává svůj vlastní, zbrusu nový, webový prohlížeč pojmenovaný Netscape Navigator, který po svém vydání okamžitě zaznamenává obrovský úspěch a brzy se stává nejrychleji prodávaným produktem v kategorii prohlížečů. Netscape rychle vydává nové a nové verze prohlížeče na svou dobu s velice atraktivním vzhledem a v roce 1995 již ovládá 80% trhu s webovými prohlížeči.

{1995 -1996}

Během tohoto nástupu mezitím vzniká malý a málo známý browser Opera, který není na rozdíl od zatím neexistujícího Internet Exploreru, ale i Netscape Navigatoru vypracován na programovém základě staříckého Mosaicu, ale vydává se svou vlastní cestou. Obrovská převaha Netscape Navigatoru v této době byla pochopitelná, protože byl skutečně jedničkou na trhu. Pak ale přichází na svět operační systém Windows 95

a společně s ním i browser pojmenovaný Internet Explorer. Do určité míry postavil Microsoft svůj byznys v oblasti webu na rozšiřování rysů HTML. Zavádí ActiveX, které udělaly prohlížeč od Microsoftu unikátním a Netscape vytvořil plug-in nazvaný Ncompass, který umožňoval ovládat ActiveX. Tato myšlenka, pomocí které jeden prohlížeč experimentuje s rozšířením HTML, aby předběhl ostatní, pokračuje dodnes.

Začíná tak válka prohlížečů. Bohužel pro Netscape má Microsoft jednu ohromnou výhodu. Jejich Internet Explorer je zdarma a navíc je automaticky přidáván do operačních systémů stejnojmenného výrobce, čímž se velice snadno a rychle rozšiřuje. Ačkoliv Netscape na rozdíl od počátečních verzí Exploreru nabízí spoustu funkcí a nástrojů zahrnujících kupříkladu podporu JavaScriptu, Javy a pluginů, Internet Explorer díky rostoucímu počtu osobních počítačů se systémy Windows, stále získával větší a větší oblast trhu, přidával nové funkce a zůstával nadále zadarmo.

{ 1996 -1997 - současnost }

IE 3.0 ze srpna 1996 již víceméně poráží Netscape v podporovaných funkcích a zejména stabilitě samotného prohlížeče. V průběhu roku 1997 je ukončena podpora pro Mosaic a na trhu tak zůstávají na výběr jediné dva větší prohlížeče. Netscape se vzhledem k rostoucímu úspěchu Microsoftu rozhoduje k vydání prohlížeče, který by byl také zdarma. Vzniká tak browser Mozilla a jeho verze jsou nabídnuty jako open source pro vývojáře celého světa. V říjnu 1997 vychází IE 4.0, který je mnohokrát pokročilejší než cokoli, co tu bylo kdy předtím. S vydáním verzí 5.0 a současné 6.0 podíl Netscapu na trhu klesl pod 30%, zatímco Explorer získával stále víc a víc. Ani Netscape verze 6.0 z listopadu 2000 již nebyla tento pokles schopna zastavit. Nemluvě o obrovském množství chyb, které tato verze obsahovala.

V současné době se do popředí zájmu začínají tvrdě drát prohlížeče Mozilla a Opera, které tvoří důstojného konkurenta masově rozšířenému Internet Exploreru. Uvidíme, zda se jim podaří dohnat či dokonce předejnat svého hlavního konkurenta, i když díky marketingové politice firmy Microsoft to bude rozhodně běh na dlouhou trať.

## 4 VÝVOJ ASP A PHP – CESTA SPRÁVNÝM SMĚREM?

### 4.1 ASP

ASP (Active Server Pages) je technologie nezávislá na programovacím jazyce (vyvinutá společností Microsoft), která umožňuje vykonávání kódu na straně serveru a následné odeslání výsledku uživateli. To znamená, že webová stránka s příponou .asp obsahuje kód, který se vykoná na IIS serveru a prohlížeči odešle pouze výsledek ve značkovacím jazyce HTML, který umí bez problému zobrazit. Programovací jazyky, které se používají, jsou tyto: VBScript a JScript.

Doby, kdy v kyberprostoru vegetovaly jenom klasické počítače, jsou nenávratně pryč. Po síti se toulá slušné množství podivných zařízení. Jak pro ně psát webové aplikace?

Doby, kdy v kyberprostoru vegetovaly jenom klasické počítače, jsou nenávratně pryč. Ledničky s vestavěným browserem sice zatím nepatří k běžnému připojení domácnosti, ale už teď se po síti toulá slušné množství podivných zařízení, počínaje mobilními telefony, konče kapesními počítači s hojným množstvím kříženců mezi nimi.

Ačkoliv název ASP.NET je odvozen od starší technologie pro vývoj webů ASP, obě technologie jsou velmi odlišné. ASP.NET je založen na CLR (Common Language Runtime), který je sdílen všemi aplikacemi postavenými na .NET Frameworku. Programátoři tak mohou realizovat své projekty v jakémkoliv jazyce podporujícím CLR, např. Visual Basic.NET, JScript.NET, C#, Managed C++, ale i mutace Perlu, Pythonu a další. Aplikace založené na ASP.NET jsou také rychlejší, neboť jsou předkompilovány do jednoho či několika málo DLL souborů, na rozdíl od ryze skriptovacích jazyků.

#### 4.1.1 Jak to celé funguje

ASP.NET ulehčuje programátorům přechod od programování klasických aplikací pro Windows do prostředí webu: stránky jsou poskládány z objektů, ovládacích prvků (Controls), které jsou protějškem ovládacích prvků ve Windows. Při tvorbě webových stránek je tedy možné používat ovládací prvky jako tlačítko (Button), nápis (Label) a další. Těmto prvkům lze přiřazovat určité vlastnosti, zachytávat na nich události, atd. Tak, jako se ovládací prvky pro Windows samy kreslí do formulářů na obrazovku, webové ovládací

prvky produkují HTML kód, který tvoří část výsledné stránky poslané do klientova prohlížeče.

První krok znamenal Mobile Internet Toolkit pro ASP.NET 1.0, verze 1.1 už má tuto funkcionalitu vestavěnou a není třeba ji instalovat zvlášť. Podporuje takzvané Mobile Web Forms. Jedná se o sadu vcelku jednoduchých ovládacích prvků (textové pole, odkaz, tlačítko...), které stačí sestavovat do formulářů. ASP.NET engine si zjistí, z jakého zařízení byl odeslán požadavek na stránku a podle toho vyrenderuje stránku jako HTML, cHTML (v našich krajích se nevyskytující "odlehčené" HTML pro mobilní telefony), WML 1.1 nebo WML 1.2, a to s přihlédnutím ke specifickým choutkám daného zařízení.

Současná verze (Device Update 4.0) obsahuje pravidla pro 265 zařízení, přičemž počet podporovaných bude několikanásobně větší (celé rodiny mobilních telefonů/PDA používají tentýž prohlížeč).

Daní za širokou kompatibilitu jsou velmi omezené možnosti ovlivnění toho, jak bude výsledek vypadat. Je třeba si uvědomit, že Mobile Web Forms fakticky podporují jenom množinu toho, co mají všechna zařízení společná. Pomocí této technologie tedy plně nevyužijete možnosti jednoho konkrétního zařízení, ale dosáhnete maximální kompatibility.

Například, pokud chcete vyvíjet složitější webovou aplikaci a víte, že vaši uživatelé (třeba obchodní zástupci) budou využívat Pocket PC, je lepší aplikaci napsat jako klasické ASP.NET stránky ve webových formátech, jenom s přihlédnutím ke specifikům kapesních počítačů. Pokud ovšem chcete, aby vaši aplikaci mohl využívat víceméně kdokoliv, můžete použít Mobile Web Forms s velkým úspěchem.

Zjednodušeně se dá říci, že v případě mobilních formulářů řadíte prvky pod sebe a nemůžete kromě pořadí příliš ovlivnit, jak přesně budou na stránce rozloženy a jak budou vypadat.

Rozlučte se tedy s myšlenkou, že napíšete jedny stránky a ty budou úspěšně fungovat a hezky vypadat na jakémkoliv zařízení. Fungovat budou, ale ovládání na běžném stolním počítači s obrazovkou obvyklých rozměrů bude působit dosti směšně. Počítejte spíše s tím, že bude stejně třeba mít dvě verze stránek, jednu pro běžné prohlížeče a jednu pro mobilní zařízení. Ono je to v konečném důsledku dobře, protože málokdy je

účelné na malém displeji kapesního počítače prezentovat stejné údaje, jako na běžných webových stránkách.

#### 4.1.2 Výhody asp.net oproti asp

Díky kompilovanému kódu běží aplikace rychleji a více chyb je zachyceno už při vývoji. Uživatelsky definované ovládací prvky lze použít jako šablony, čímž se významně redukuje duplicitní kód. Podobný přístup jako k aplikacím pro Windows zjednodušuje přechod od jednoho prostředí k druhému. Bohatý výběr ovládacích prvků a knihoven tříd velmi zrychluje vývoj aplikací. Programátoři mají na výběr velké množství programovacích jazyků. Schopnost cachovat celou stránku nebo pouze její části podstatně zvyšuje výkon serveru. Lze jej provozovat na různých operačních systémech i webových serverech. Počínaje verzí 2 generuje ASP.NET validní HTML 4.0 / XHTML 1.0 / XHTML 1.1 kód a JavaScript. Stavové prostředí nad bezstavovým protokolem.

Ačkoliv webový protokol HTTP je sám o sobě bezstavový, událostmi řízené programování zachování stavu (uchování kontextu mezi jednotlivými požadavky) vyžaduje. ASP.NET tento problém řeší kombinací HTML a JavaScriptu pomocí dvou základních technik:

- *ViewState* uchovává informace mezi postbacky (opakováním odesíláním formuláře na server) v zakódovaném tvaru ve skrytých formulářových polích. Jeho výhodou je, že využívá pouze HTML a nevyžaduje žádnou speciální podporu na straně serveru ani klienta. Nevýhodou je, že se mezi serverem a klientem přenáší větší objem dat, zejména je-li *ViewState* využíváno nesprávně.
- *Session State* oproti tomu ukládá veškeré informace na straně serveru a předává (typicky jako cookie nebo součást URL) pouze jednoznačný identifikátor. To sice zmenšuje objem přenášených dat, ale klade vyšší nároky na výkon serveru. Pokud se *sessions* používají nesprávně, může být server náchylný i k Denial of Service útokům. Oproti ASP umožňuje ASP.NET ukládání *session state* do samostatného procesu nebo na SQL server. To zjednodušuje použití *session* ve webových farmách, zvyšuje výkon a umožňuje stav zachovat i při restartu serveru.



## 4.2 PHP

PHP se stalo jedním z nejpopulárnějších systémů pro tvorbu dynamických webových stránek. Proto každá jeho nová verze vzbuzuje vlnu očekávání. Nejinak tomu bylo i před vydáním PHP5. V tomto článku bych chtěl shrnout své nepříliš pozitivní dojmy z této novinky a směřování platformy PHP vůbec.

Jazyk PHP se podle mě začal ubírat směrem, který mu rozhodně neprospěje. Jeho tvůrci se snažili maximálně vylepšit objektový model jazyka, který nemůže být nikdy tak použitelný jako u Javy nebo C#, protože je silně limitován netypovostí PHP, ale zapomněli přitom na jiné problémy, které by potřebovaly rychlé a úplné řešení. Místo, aby posílili PHP tam, kde je skutečně výborně použitelné (malé a střední aplikace s požadavkem na rychlost), marně se snaží konkurovat robustním platformám, které jsou však určeny na úplně jiný typ aplikací než PHP.

Systém PHP se od své první verze s primitivní funkcionalitou vyvinul v mocný programovací nástroj s masivní oblibou a podporou vývojářů. Jeho obliba je natolik vysoká, že je považován svým způsobem za standard a málokterý hosting (včetně freehostingu) jej nepodporuje. V poslední době se zvednul velký povyk okolo nejnovější verze PHP s pořadovým číslem 5. Nebudu se zabývat změnami, které tato novinka přinesla (na Intervalu si o tom můžete přečíst třeba v článku [Objekty v PHP5](#) nebo [Další novinky v PHP5](#)), ale pokusím se zde podat argumenty, proč pátou verzi PHP nepovažuji za krok správným směrem.

### 4.2.1 Směřování PHP

Dobrou vlastností každého vývojáře je mimo jiné i správná volba prostředků. Úměrně velikosti celého projektu a některým dalším požadavkům musí být zvolena správná programovací platforma a jazyk, protože špatná volba v této fázi vývoje může velmi negativně ovlivnit celý proces a hlavně náklady, o které jde především. Proto je například nevhodné použít pro malý web pracující nad nevelkou databází řešení založené na .NET, protože by to bylo ono pověstné "braní kanónu na vrabce". Zcela analogicky pro rozsáhlé robustní projekty není PHP tím nejlepším řešením, ale nabízí se zde například právě zmiňovaný, .NET nebo řešení založená na Javě.

Základním problémem PHP jsou nesplnitelné ambice jeho tvůrců a bohužel i jeho některých "tvrdých" fanoušků. Místo toho, aby si PHP upevňovalo svou vybudovanou

pozici na poli malých až středních aplikací a opatrně zvažovalo každou úpravu, snaží se nesmyslně proniknout výše, kde však již kralují úplně jiná řešení. Podobně nesmyslné, až úsměvné mi připadají názory některých jedinců, že PHP stačí na jakýkoliv projekt. Výsledkem těchto snah však nemůže být nic jiného než něco, co už přestává být úplně použitelné pro rychlý a efektivní vývoj menších aplikací, a co ještě zdaleka není použitelné pro robustní systémy.

Zcela v duchu těchto ambicí byla vytvářena pátá verze jazyka PHP. Na úkor oprav mnohých chyb a nedostatků, kterými současné PHP dost trpí, byl značně rozšířen objektový model. Ovšem každý stavař vám řekne, že na nestabilních základech nelze postavit kvalitní stavbu. Podobný problém podle mě postihnul právě PHP.

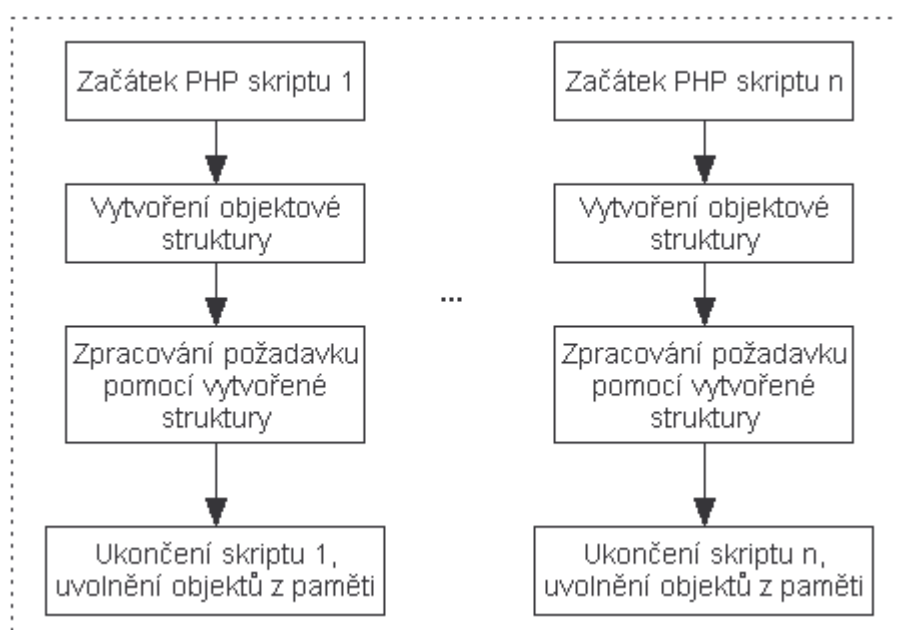
#### 4.2.2 Rozšíření objektového modelu

Jak již bylo řečeno, největších změn se dočkal objektový model. Přibyly modifikátory přístupu k atributům a metodám třídy (public, protected, private), možnost napsat vlastní destruktory, lepší podpora kopírování objektů a některé další možnosti, nebo které mají PHP přiblížit objektovým jazykům jako je Java.

Jednou ze základních vlastností PHP je netypovost. Ta umožňuje použití specifických prostředků, které nemůže poskytnout jazyk s typovou kontrolou a naopak. Realizace objektových postupů v takovém jazyce je však značně limitována právě jeho netypovostí. Nedosáhneme například nikdy skutečného polymorfismu, protože metodám nelze předepsat typ parametru, který mají očekávat.

Objektově orientované programování má sice v PHP svůj význam, ale ten by měl být odvozován od skutečného určení tohoto jazyka. Objekty je určitě vhodné použít pro lepší organizaci zdrojového textu. Například většina z programátorů si časem vytvoří třídu pro zasílání dotazů na databázi (Zjednodušte si práci s MySQL). Taková třída obvykle zapouzdřuje nejen funkce PHP, ale i proměnné představující identifikátor připojení a výsledku dotazu jako své atributy. Tyto proměnné pak zbytečně neznečišťují globální názvový prostor, ale jsou zapouzdřeny v dané instanci takové třídy. Proto nemusíme při programování dávat pozor na to, že v proměnné `$link` máme uložen identifikátor připojení, a že jej nesmíme přepsat. Toto základní využití objektů rozhodně považuji za správné a především výhodné. Zároveň se však domnívám, že zde někde se nachází strop využití objektového programování v PHP.

OOB přineslo zcela nový pohled na koncepci programování. Tou se stala množina objektů (instancí tříd) specializovaných na řešení určitého problému, které spolu komunikují a implementují tak požadovanou funkčnost. Jejich životnost je různá, některé jsou použity pro vykonání určité činnosti a pak hned zrušeny, jiné žijí po celou dobu běhu aplikace. Lze tedy říci, že doba života objektů v paměti je shora omezena pouze dobou běhu aplikace. Životnost objektu v PHP je však mnohem limitovanější! Objekt existuje pouze v průběhu jediného požadavku.



Obr. 4. Implementace objektových postupů v PHP

Při každém novém požadavku se musí vytvářet celá objektová struktura a definovat stavy jednotlivých objektů. Navíc, pokud jsou některé třídy součástí nějaké hierarchie (dědičnost, implementace), musí skript načíst a zpracovat celou tuto hierarchii. A to vše pro splnění značně omezeného úkolu, pro který je daný skript určen. Při dokončení skriptu mizí všechny tyto objekty v nenávratnu.

Určitou berličkou při řešení této situace by bylo použití serializace celé objektové struktury a její deserializace na začátku skriptu následujícího (na Intervalu se o ní můžete více dočíst ve článku PHP pro pokročilé - znovu třídy a objekty). Pak je ovšem vhodné se zamyslet nad použitím úplně jiné platformy, která lépe splňuje naše požadavky.

Cílem každého programátora píšícího v PHP by měl být skript, který co nejefektivněji vyřeší daný problém. Na základě výše uvedených argumentů si dovoluji

tvrdit, že přílišný objektový přístup tomu zamezuje. Proto se mi nezdá správné, že vylepšení podpory objektů v PHP5 bylo věnováno nejvíce prostoru. Přitom mnohé problémy, které sužují PHP mnohem palčivěji, zůstaly nevyřešeny. A navíc přidáním objektové podpory vznikly některé další.

#### 4.2.3 Podpora různých kódování řetězců

Tohle je myslím nejpalčivější bolest PHP. Zatímco všechny moderní jazyky a platformy se snaží poskytnout co největší podporu pro různá kódování řetězců, PHP se staví do pozice mrtvého brouka. Pro vývojáře, kteří chtějí své aplikace používat v různých prostředích s různými jazyky, tak vyvstává skutečný problém, především v tu chvíli, kdy chtějí použít vícebytové kódování. Všechny standardní funkce PHP počítají s tím, že znak je kódován pomocí jediného bytu, což však nemusí být vždy pravda. PHP5 se tento problém snaží vyřešit pomocí knihovny iconv a jejích funkcí, která je nově přímou součástí PHP. Do skutečné nativní podpory různých kódování má však tato knihovna daleko.

#### 4.2.4 Dodržování zásad při pojmenovávání

Vizitkou každého programátora je jeho kód. Nutnou podmínkou pro čistotu kódu je dodržování pojmenovacích konvencí pro třídy, metody a proměnné. Vývojáři PHP však stále přešlapují na místě a nejsou schopni dodržovat konvence ani na úrovni nativních funkcí. A tak můžeme i v páté verzi narazit třeba na funkci `stripslashes()` (všechno malým písmenem) a hned vedle ní na `strip_tags()` (slova\_oddělená\_mezerníkem). Nové zmatení přichází s větší podporou objektového přístupu. A tak zde máme třídu `mysqli` (opět všechno malým písmenem) pro zapouzdření připojení k databázi, zatímco třídy ze standardní knihovny SPL se již drží klasické konvence Všechna Slova Velkým Písmenem.

Tento chaos přetrvává v PHP již velmi dlouhou dobu a každá nová verze jej pouze prohlubuje. Osobně bych byl všemi deseti pro zažitou javovskou konvenci (každé slovo identifikátoru velkým písmenem, první písmeno velké u názvu třídy, jinak malé), ale nevadilo by mi, kdyby byla dodržována alespoň nějaká konvence!

#### 4.2.5 Chaotická implementace výjimek

Další avizovanou novinkou v PHP5 je podpora výjimek známých z Javy nebo C++. Bohužel je v duchu celé pětkové verze nedotažená. PHP totiž neposkytuje hierarchii výjimek podobně jako moderní objektové jazyky, ale poskytuje pouze základní výjimku `Exception`. Navíc vzhledem k netypovosti PHP můžete pomocí klíčového slova `throw` vyhodit jakýkoli objekt, který s touto třídou nemusí mít absolutně nic společného.

Toto však není ten největší problém výjimek v PHP5. Mnohem horší je jejich podpora nativními funkcemi PHP. Je totiž nulová! Například bych naivně předpokládal, že funkce `mysql_query` při zadání syntakticky špatného dotazu vyhodí této situaci ekvivalentní výjimku, třeba `SQLException`. Místo toho však tato funkce pouze vrátí hodnotu `false`. Zcela stejně tomu je u zbývajících funkcí.

Nabízí se tedy otázka, proč vlastně tvůrci implementovali možnost výjimek, když je pak sami nepodporují.

#### 4.2.6 Ztracená příležitost

Jazyk PHP se podle mě začal ubírat směrem, který mu rozhodně neprospěje. Jeho tvůrci se snažili maximálně vylepšit objektový model jazyka, který nemůže být nikdy tak použitelný jako u Javy nebo C#, protože je silně limitován netypovostí PHP, ale zapomněli přitom na jiné problémy, které by potřebovaly rychlé a úplné řešení. Místo aby posílili PHP tam, kde je skutečně výborně použitelné (malé a střední aplikace s požadavkem na rychlost), marně se snaží konkurovat robustním platformám, které jsou však určeny na úplně jiný typ aplikací než PHP.

PHP je dnes jediným prakticky použitelným řešením pro menší aplikace, a proto jej budu k těmto účelům nadále používat. Pro velké aplikace, u kterých je objektový přístup dnes povinností, je však stále nutno volit jinou platformu, bez ohledu na novinky v páté verzi PHP.

## **II. PRAKTICKÁ ČÁST**

## 5 PŘÍKLADY WEBOVÝCH PROHLÍŽEČŮ

### 5.1 Grafické webové prohlížeče

Internet Explorer

Mozilla

Opera

Netscape Navigator

Beonex

Firebird (dříve Phoenix)

IBM (webový prohlížeč pro OS/2)

Aphrodite

Galeon pro Gnome

Salamander

Epiphany

Skipstone

K-Meleon pro Windows

Camino pro Macintosh OS X(Chimera)

Oregano

Konqueror(zobrazoval Khtml)

Abrowse

Safari

SkyKruzer

Amaya

iCab

OmniWeb

Dillo

IBrowse

A Web

Voyager

Espial Escape

HotJava

Arachne

Off By One

Emacs/W3

## **5.2 Textové webové prohlížeče**

Links, Lynx, w3m, Netrik

### **Dřívější prohlížeče, které se již nevyvíjejí**

Arena, Cello, CyberDog, MidasWWW, Mosaic, Viola, WorldWideWeb

## **5.3 Přehled verzí nepoužívanějších prohlížečů**

### **Internet Explorer**

1.0 Srpen 1995

2.0 Listopad 1995

3.0 Srpen 1996

4.0 Říjen 1997

5.0 Březen 1999

5.5 Červenec 2000

6.0 Říjen 2001

### **Macintosh IE**

2.0 Duben 1996

2.1 Září 1996

3.0 Leden 1997



4.0 Leden 1998

4.5 Leden 1999

5.0 Březen 2000

### **Mosaic**

1.0 Listopad 1993

2.0 Listopad 1995

2.1 Leden 1996

3.0 Leden 1997

### **Netscape**

1.0 Prosinec 1994

1.1 Duben 1995

2.0 Březen 1996

3.0 Srpen 1996

4.0 Leden 1997

4.5 Srpen 1998

6.0 Listopad 2000

7.0 Srpen 2002

### **Opera**

2.1 Prosinec 1996

3.0 Prosinec 1997

3.5 Listopad 1998

4.0 Červen 2000

5.0 Prosinec 2000

6.0 Listopad 2001

7.0 Leden 2003

### **Mozilla**

0.6 Říjen 2000

1.1 Srpen 2001

1.6 Říjen 2003

## 6 SOUČASNÉ WEBOVÉ TECHNOLOGIE

### 6.1 Webové prohlížeče

#### 6.1.1 Opera

Webový prohlížeč Opera byl vytvořen jako alternativa k populárnímu Internet Exploreru a Netscape Navigatoru norskou společností Opera Software. Je dostupná pro systémy BeOS, Linux, FreeBSD, Solaris, MacOS, OS/2, QNX, Symbian OS a všechny 16ti i 32ti bitové verze Microsoft Windows. V prosinci 2000 přešla Opera z demoverze na 30ti denní trial verzi a nabídla bezplatné stažení z webu s tím, že se zobrazují v okně prohlížeče reklamní nabídky. Zdrojové kódy aplikace nejsou dostupné pro koncové uživatele.

Opera se stala známou pro rozhraní MDI, které zobrazuje veškeré otevřené stránky pomocí záložek v jednom původním rodičovském okně. Toto rozhraní je již zahrnuto například i u nadstavby Internet Exploreru MyIE2. Nejnovější verzí je Opera 7 beta release uveřejněná v listopadu 2002, která obsahuje nový renderovací model, nový uživatelský interface s možností použití skinů a také nový mail a news klient nazvaný M2.

#### 6.1.2 Mozilla

Mozilla je bezplatný softwarový balík, jehož komponenty zahrnují webový prohlížeč a e-mailový klient. V únoru 1998 firma Netscape uvolnila zdrojové kódy Netscape Communicatoru, který také obsahoval prohlížeč Navigator. Na podporu dalšího rozvoje vývoje Mozilly firma založila Mozilla organization a webovou stránku mozilla.org. Nicméně vývoj na základě kódu Communicatoru byl obtížnější než se z počátku zdálo. Z toho důvodu bylo rozhodnuto, že vývojáři Mozilly nebudou používat starý kód, ale vytvoří kompletně nové rozhraní a HTML engine. Tak vznikl Gecko HTML layout engine.

Mozilla dokáže pracovat pod operačními systémy Linux, MacOS, Solaris a Windows. Přinesla kromě jiného podporu pro nastavování uživatelského soukromí a bezpečnosti. Silně podporuje standardy W3 konzorcia. Současné verze Mozilly umožňují vysokou nastavitelnost funkcí a podporují pokročilé možnosti správy obrázků, hesel,

popup oken, cookies a prohlížení přes záložky v jediném okně. Nejnovější verzí je rychlá a stabilní verze 1.6.

### 6.1.3 Internet Explorer

Je to známý webový prohlížeč od firmy Microsoft, velice oblíbený prohlížeč a naprosto dominantní na současném trhu zejména kvůli faktu, že je zdarma přidáván k operačním systémům Windows a instalován na OS X firmy Apple Macintosh. Pro Internet Explorer bylo a stále bývá vydáváno mnoho bezpečnostních záplat, což ukazuje na nejmenší zabezpečení vůči útokům zvenčí z těch několika málo hlavních prohlížečů. Renderovací modul HTML pro Windows verzi Exploreru je využíván jako interface pro několik dalších prohlížečích aplikací jako jsou MyIE2, Crazy browser, NetCaptor nebo NeoPlanet. Podle nejnovějších zpráv a také mezivýsledků nejrůznějších soudních přelíčení se dá předpokládat, že Internet Explorer bude již brzy existovat jako samostatná aplikace a ne jako součást operačního systému Windows.

### 6.1.4 Netscape Navigator

Prohlížeč s velmi dobrou výchozí pozicí na trhu, neboť patřil vůbec k jedněm z prvních. Je založen na kódu prohlížeče Mosaic. Také toho dokázal náležitě využít, zavedl rámy (verze 2.0), cookies, JavaScript a v jednu chvíli ovládal téměř 90% veškerého trhu s prohlížeči. Nicméně, díky činnosti Microsoftu a jeho bezplatnému Internet Exploreru včleňovanému přímo do operačních systémů postupně ztrácel své pozice a v roce 1998 byl jeho vývoj ukončen. Nástupcem Netscape Navigatoru se stala Mozilla, která funguje dodnes.

### 6.1.5 Kritika

Nejčastěji vznášená kritika na výše zmiňované prohlížeče se nejvíce dotýká Internet Exploreru a Netscape.

Webový prohlížeč Internet Explorer je často kritizován (verze 6.0 či starší) pro svou bezpečnostní politiku, která umožňuje šíření virů, spyware či adware. Chyby v tomto prohlížeči bývají často zneužívány k získávání kontroly nad systémem či získávání citlivých údajů o uživatelích. Tvůrcům prohlížeče se též často vyčítá malá podpora webových standardů a jejich častá chybná implementace. Řada vývojářů tvrdí, že tím

společnost Microsoft brzdí rozvoj v oblasti webových technologií. Aktuální verze (7.0) přináší v této oblasti určité zlepšení.

Tab. 2. Hodnocení prohlížečů

	Web	Java	Pop-up	Stahování	Banky
Firefox	4	5	4	4	3
Internet Explorer	5	5	5	2	5
Mozilla	4	5	5	3	3
MyIE	5	5	5	2	5
Opera	4	3	4	4	3

Webový prohlížeč Netscape navigátor byl většinou kritizován za splnění nestandardního HTML značkovacího rozšíření, které se někdy odkazovalo na symbol pro Netscape. Netscape byl také kritizován za následující aktuální webové standardy, často se opožďoval nebo je podporoval nesprávně.

## 6.2 Co hýbe dnešními webovými stránkami

Interaktivní aplikace jsou velmi důležitou součástí Internetu. Asi každý dnes zná Flash, ale ten není zdaleka jedinou možností, jak oživit webové stránky. V několika článcích se proto podíváme na současnost i budoucnost těchto technologií a popíšeme ty nejznámější z nich.

Zkratka RIA označuje sousloví Rich Internet Applications, a jak již název napovídá, jedná se o webové aplikace, které umožňují „bohatší“ zážitek ze surfování kyberprostorem. Obecně se tím myslí zejména interaktivita, jejíž příchod proměnil pasivní internetové stránky ve webové aplikace, které na uživatele reagují, je možné je ovládat a poskytují zpětnou vazbu. Ve srovnání s klasickými webovými aplikacemi (do kterých by se dal zařadit zrovna třeba nějaký ten chat) už složitější RIA nevystačí pouze s kódem HTML a elementární komunikací klient – server; ke své práci obvykle potřebují nějaký dodatečný program, přehrávač či plugin. Mezi nejznámější takové „zprostředkovatele interaktivity“ patří zejména Flash nebo Java – a samozřejmě i další, na které se podíváme v následujících odstavcích.

### 6.2.1 Java & JavaScript

I když zmíněné názvy vypadají podobně, nejedná se o totéž. Java je objektově orientovaný programovací jazyk a jde o technologii, která vznikla jako interní projekt společnosti Sun Microsystems už někdy kolem roku 1990. Na Internet se sice dostala až v roce 1995 (kdy byla představena), ale i tak se jedná o jednu z prvních technologií, jak vytvářet RIA. Programovací jazyk Javy vychází z C a C++, ale je o něco jednodušší. Díky speciálnímu softwaru na straně klienta (Java Virtual Machine) umožňuje běh i relativně složitých aplikací v rámci prohlížeče (ve formě tzv. Java appletů). Velikou výhodou Javy je také schopnost fungovat napříč platformami (Windows, Mac OS, Linux).

JavaScript je známý skriptovací jazyk, který se objevil na přelomu let 1995 a 1996 jako součást prohlížeče Netscape Navigator. Jeho možnosti jsou skutečně široké – od otvírání nových (pop-up) oken až po šikovné funkce webových stránek. Na bázi JavaScriptu pracuje například Gmail, který umožňuje průběžnou automatickou aktualizaci doručené pošty, a současně lze třeba chatovat s jinými připojenými uživateli.

V poslední době se lze také často setkat s označením „AJAX“, což je v podstatě kombinace klasických webových technologií jako html/xml právě s JavaScriptem (však také zkratka AJAX označuje „Asynchronous JavaScript and XML“). Ačkoliv se AJAX často označuje jako nový přístup v oblasti webových aplikací (zejména ve spojení s obratem „Web 2.0“), jedná se ve skutečnosti o balík „starých známých“ prvků. Takže když jsem zmínil, že Gmail využívá JavaScript, tak lze zároveň říci, že Gmail pracuje s technologií AJAX, protože se jedná v podstatě o to samé.

### 6.2.2 Kreativita jménem Flash

Flashové technologie se na Internetu objevily zhruba ve stejném období jako Java, tedy v roce 1996. Flash se v samotných počátcích ovšem nezaměřoval na interaktivitu jako takovou, ale zejména na oblast grafiky a vektorových animací. Nicméně v průběhu let se vyvinul ve velice schopný multimediální kreativní nástroj s téměř neomezenými možnostmi a schopnostmi. V současnosti tak v podstatě dominuje oblasti interaktivní grafiky na World Wide Webu a je možné narazit i na celé webové stránky vytvořené právě ve Flashi.

Co se historie týče, vše začalo před jedenácti lety, když společnost Macromedia odkoupila program FutureSplash Animator firmy FutureWave Software. Světlo světa tak spatřil Flash 1 a skoro každý další rok pak vycházely nové verze, které se neustále zdokonalovaly a vyvíjely (kupříkladu hned druhá verze z roku 1997 představila tlačítka, tedy první krok k nějaké základní interakci). Na konci roku 2005 byla společnost Macromedia pohlcena ještě větším gigantem, společností Adobe Systems, pod jejíž hlavičkou vyšla před časem zatím poslední verze, Adobe Flash CS3 (= Creative Suite 3).

V současné době jsou flashové technologie schopné téměř všeho. Na nejzákladnější úrovni se s Flashem můžeme setkat ve formě reklamních bannerů či zábavných animací, ale to je jen ona pomyslná špička ledovce. Součástí Flashe je totiž programovací jazyk ActionScript, s jehož pomocí lze vytvářet interaktivní aplikace nebo třeba i jednodušší hry. Flash dokáže pracovat s textem, bitmapovou i vektorovou grafikou a dalšími multimédii; rozumí si se vzdálenými databázemi a velmi často se používá třeba na streamování videa (viz např. YouTube).

### **6.2.3 Interaktivita sem, interaktivita tam...**

Jak je z předchozího textu patrné, je z čeho si vybírat. V současné době existuje celá řada možností, jak vytvořit interaktivní aplikace (ať už v rámci okna webového browseru, či jako desktopovou aplikaci komunikující se vzdálenými servery). Nepříjemností je ale ne vždy dostatečná kompatibilita (nejen mezi systémy a browsery, ale třeba i mezi různými verzemi stejného frameworku). Můžeme si povzdychnout, že například W3C poněkud zaspalo dobu, ale to je tak všechno, co s tím asi můžeme dělat.

Ovšem, aby toho náhodou nebylo málo, do stávajícího „boje o interaktivitu“ se nezapojují pouze nynější hráči (třeba s novými verzemi stávajících technologií), ale objevují se i řešení nová. Microsoft SilverLight, Adobe AIR, JavaFX – to jsou asi nejdůležitější příklady nových konceptů.

## 7 BEZPEČNOST WEBU

Tato část je zaměřena na bezpečnostní problematiku související s technologiemi implementovanými ve webových prohlížečích - jedná o:

- Javascript
- Cookies
- SessionID

Kromě těchto rizik jsou další rizika spočívající ve slabinách serverových technologií jako je serverový operační systém (Linux, Windows, freeBSD), webový server (Apache, IIS), serverový skriptovací engine (PHP, ASP, ASP.NET, Java, ...) a databáze (MySQL, MS SQL Server). Bezpečnostní problematika těchto technologií přesahuje rámec tohoto článku, zmiňujeme zde jenom všeobecně známý problém průniku přes SQL dotazy v databázích:

- SQL injection

### 7.1 Javascript

#### 7.1.1 Bezpečnostní rizika Javascriptu

Javascript je dnes (r. 2007) implementován ve všech moderních webových prohlížečích. Jeho implementace do webových prohlížečů byla motivována snahou výrobců webových prohlížečů získat konkurenční výhody oproti ostatním. Javascript má dva rysy, které z něj činí ideální nástroj k prolomení bezpečnosti webových aplikací:

- Kód Javascriptu lze k HTML-dokumentu připojit odkazem LINK v záhlaví dokumentu, proti tomu nelze nic namítnout, ale další možnosti jsou velmi nebezpečné - Javascript lze vložit jako blok kódu (tag SCRIPT) kamkoliv do těla HTML-dokumentu, kde by měl být pouze značkovaný obsah, lze jej dokonce vložit jako hodnotu atributů ovladačů událostí kteréhokoliv z běžných HTML-elementů (atributy onclick, onload, onmouseover,...)
- Javascript má ničím neomezený přístup k jakýmkoliv datům uloženým v HTML-dokumentu, může neomezeně manipulovat obsahem stránky, ubírat, měnit a přidávat jakýkoliv element ve stránce, přesměrovat stránku, odečíst hodnoty cookies s citlivými informacemi, zavést do prohlížeče jiný skript z kteréhokoliv



serveru, spustit soubor Java nebo ActiveX, které mají zase řádově větší ničivý potenciál než samotný Javascript

Z důvodu popsaných bezpečnostních rizik má Javascript implementovaný ve webových prohlížečích řadu funkčních omezení. Přes tato omezení je stále Javascript vložený do webové stránky výborným nástrojem pro potenciálního útočníka. Abychom eliminovali všechna bezpečnostní rizika, musíme dobře znát výše uvedené dvě skupiny otázek, které raději zopakujeme:

- všechny možnosti, jak může útočník vložit svůj javascript do cizích webových stránek
- všechny možnosti, jak může kód javascriptu ohrozit bezpečnost webové aplikace

**Důležitá poznámka:** IE má implementován vedle Javascriptu i jiné podobné technologie (VBScript), které může útočník využít stejně jako Javascript.

Poznamenejme, že Javascript nepřináší k základní funkcionalitě HTML(XHTML) žádná nová podstatná vylepšení a 90% všech webových aplikací může bez problémů fungovat bez Javascriptu. Webový prohlížeč bez Javascriptu a bez pluginů pouze jako HTMLparser, je pro hackera nedobytnou pevností.

Javascript může být do stránky vložen takto:

- Přímo jako tag `<script src="http://www.utocnik.cz/script.js" ></script>` zadaného do nesprávně ošetřeného textového formulářového prvku. Takto vložený skript buďto přímo obsahuje škodlivý kód, nebo si kód stáhne v externím souboru (atribut src) Tento skript se spustí buďto ihned po vložení, nebo na základě nějaké události (odeslání formuláře, klik na některý odkaz apod...).
- IFRAME s cizím obsahem. Takovýto objekt může dělat vše, co se netýká původní stránky, tedy otvírat nová okna, přesměrovat prohlížeč apod.
- IFRAME s vlastním obsahem. Jelikož jde o stejnou doménu, má díky filozofii DOM stejný ničivý potenciál , jako skript vložený pomocí tagu `<script>` .

Javascript může provádět řadu věcí, např.:

- přesměrovat stránku na jinou URL-adresu
- vložit do stránky nové objekty (např. formulář vedoucí na server útočníka)

- pozměnit vzhled a polohu. stávajících elementů ve stránce (nebo je zcela ze stránky vyjmout)
- odečítat veškeré hodnoty zadávané přes klávesnici
- blokovat kliknutí na jakýkoliv odkaz
- přesměrovat kliknutí na odkaz na jinou url - adresu
- stáhnout a spustit program napsaný v Java
- stáhnout a spustit ActiveX komponentu

I to nejmenší - pouhé přesměrování stránky může způsobit provozovateli webového serveru ekonomickou ztrátu. Webový server přestane být funkční a cílová adresa přesměrování může být url nekalé konkurence.

### 7.1.2 Cross-frame scripting

Hierarchická struktura DOM, v níž je Javascript implementován, v sobě skrývá specifické bezpečnostní riziko spočívající v možnosti přistupovat např. z jednoho rámu k obsahu jiného. Úplně nahoře ve struktuře DOM je objekt window, pod který spadají jednotlivé frames (rámy) se svými dokumenty. Z filozofie takového hierarchického členění je možné z jednoho rámu přistupovat k obsahu rámu druhého, což s sebou nese jistá bezpečnostní rizika. Shrňme si nyní, jaká jsou práva jednotlivých ráků a kam můžou přistupovat.

Dokumenty k sobě mohou navzájem přistupovat, pouze pokud se shodují jejich „doménové“ URL. V podstatě by zde mohlo být uvedeno totéž co u cookies, včetně výjimek u některých hojně používaných domén druhého stupně. Navíc HTTP obsah nemůže přistupovat k HTTPS obsahu (Secure Socket Layer). Omezení jsou zde i v případě čtení cizí URL, což by se útočníkovi mohlo případně hodit ke čtení SessionID z aktuální adresy. V tabulce vidíte možnosti manipulování s adresami jednotlivých ráků.

Možnosti manipulování s parametrem SRC (tj. URL) framu cizí domény:

- window.location - vlastnost může být nastavována pro navigování ale ne čtena
- Ostatní funkce location - jsou zablokovány
- document.href - vlastnost může být nastavována pro navigování ale ne čtena

- ostatní funkce document - jsou zablokovány
- <iframe> src - vlastnost může být nastavována pro navigování, ale ne čtena

## 7.2 Cookies

### 7.2.1 Bezpečnost cookies a chyby webových prohlížečů

Cookies je informace, kterou na klientský počítač ukládá webový server (rep. navštívená webová stránka) za účelem identifikace uživatele a pro statistická měření. Aby server nemohl ohrozit bezpečnost klientského počítače, je funkcionality cookies výrazně omezena. Každý server může uložit do klientského počítače maximálně 20 cookies o délce 4 kB. Číst informace uložené v cookies by měl mít možnost pouze ten server, který je tam uložil - to znamená, že by nikdy neměl nastat případ, že by server mohl přečíst cizí cookies. Práva ke čtení cookies jsou odvozena od url-adresy webového serveru. Práva ke čtení cookies jsou vázána na určitou URL - obvykle doménu druhého stupně (server.cz), ale mohou být vázána s jakýmkoliv adresářem či dokumentem.

Zamezení čtení cookie jiným serverem než tím, který je uložil, je naprosto zásadní bezpečnostní otázka, která závisí na kvalitní implementaci této zásady v jednotlivých prohlížečích. V minulosti se vyskytly v různých prohlížečích bezpečnostní díry v této oblasti, z nichž uvedme pro ilustraci známý případ Internet Exploreru.

### 7.2.2 Ztráta anonymity měřením návštěvnosti webových stránek

Má-li uživatel při pohybu po webových stránkách povoleno ukládání cookies ve svém prohlížeči, umožní statistickým serverům, které monitorují návštěvnost webových stránek a chování uživatelů na internetu, aby o jeho pohybu vedly přesné záznamy, jaké stránky navštívil a jak tam byl dlouho, popř. co přesně si na stránkách prohlížel. Tyto statistické servery pak nabízejí získané a analyzované podrobné statistiky o návštěvnosti webových stránek a chování uživatelů na internetu zájemcům pro jejich marketingové strategie. Servery ukládají permanentní cookies návštěvníkům měřených serverů při návštěvě každé stránky (tyto praktiky jsou běžné u statistických webových služeb jako je Navrcholu nebo Toplist).

Měření návštěvnosti předpokládá vložení malého HTML kódu do měřených webových stránek. Je několik mechanismů, jak by se potenciální útočník mohl

k informacím, které cookies ze serveru Navrcholu.cz obsahují dostat. Jednou z nich je případná bezpečnostní díra přímo na serveru Navrcholu.cz, jinou možností je bezpečnostní díra ve vyhledávači kteréhokoliv z návštěvníků měřených serverů, kterou může využít webová stránka útočnicka proti svým návštěvníkům.

Pokud dovolíme, aby útočník vkládal na naše stránky svůj kód, otevíráme mu čistě teoreticky další možnost průniku. Jelikož je takovýto cizí kód považován za náš, cross-frame scripting policy popisovaný o pár odstavců níže ho povolí a obsah naší stránky je na úrovni DOM k dispozici útočnickovi.

## 7.3 Session

### 7.3.1 Prozrazení sessionID

Session je záznam na serveru, který se inicializuje při našem načtení první stránky a skončí buď vypršením určitého časového intervalu, nebo pokud přímo klikneme na odkaz "odhlásit se", typicky v nějaké webové aplikaci. Hlavní výhodou je, že od té doby si nás server vede ve své interní databázi a programátor může používat proměnné v rámci této session.

Účel session je dvojit:

- umožní webové aplikaci uchovat informaci mezi jednotlivými stránkami o provedených akcích uživatele (něco jako "trasu")
- umožní ověřit identitu, kterou na začátku získá logovacím procesem (zadáním hesla) a zpřístupnit neveřejný obsah webové aplikace pouze oprávněným uživatelům

Typický logovací mechanismus funguje takto. Při prvním vstupu na webovou stránku se uživateli založí tzv. session a odešle se mu identifikační cookie, která obsahuje jednoznačný náhodně vygenerovaný identifikátor, tzv. sessionID. Toto sessionID je uloženo také na serverové straně. Při každém dalším vstupu na webové stránky tohoto serveru odesílá webový prohlížeč v HTTP-hlavičce tuto cookie a server porovná tuto hodnotu s hodnotami, které si uchovává. Pokud nalezne odpovídající session, je uživatel identifikován a webová aplikace si může v session přečíst o uživateli různé trvalé informace. Jednou z nich je proměnná indikující, zda se uživatel prokázal platnými přihlašovacími údaji (heslo, uživatelské jméno, apod...), tato proměnná se jmenuje obvykle

logged a na začátku je nastavena na FALSE. Po odeslání platných přihlašovacích údajů je tato proměnná nastavena na TRUE a obvykle je vstup na neveřejné stránky serveru řízen právě touto proměnnou.

### 7.3.2 Metody předávání sessionID

Jsou tři technické způsoby, jak udržet vlákno session:

- HTTP autentizace
- přenos sessionID v url-adrese (sessionID je uvedeno v každém odkazu v parametrické části URL)
- přenos sessionID mechanismem cookie

Pro všechny tři způsoby platí, že sessionID odeslané prohlížečem v HTTP-dotazu jsou serverovým skriptem ověřeny, zda se shodují s hodnotou uloženou na serveru. SessionID získá uživatel na základě odeslání správných logovacích údajů (heslo, uživatelské jméno).

HTTP autentizace využívá logovacího dialogu integrovaného přímo v prohlížeči. Při žádosti o neveřejně přístupnou stránku (např. kliknutím na odkaz) se nám otevře dialog se jménem a heslem. Zadané heslo a jméno je odesláno na server, který zkontroluje platnost údajů a nejsou-li údaje platné, trvá na zadání správných údajů. Po zadání správných přihlašovacích údajů jsou tyto údaje zapamatovány v prohlížeči (nikoliv jako cookie) a jsou prohlížečem dále odesílány v každém dalším požadavku v HTTP hlavičce. Výhodou je, že tyto údaje nelze odposlouchat Javascriptem a nejsou ani později přístupné pomocí Javascriptu. Nevýhodou je malý komfort při ošetření dialogu zadávání uživatelského hesla.

Přenos sessionID pomocí cookies je nejpohodlnější a nejpoužívanější způsob. Při přihlášení heslem vygeneruje serverový skript unikátní identifikátor daného sezení - tzv. sessionID, které odešle klientovi pomocí cookie. Server si toto sessionID zapamatuje a současně je odešle pomocí cookie a uloží na klientský počítač. Při každém požadavku na novou stránku odesílá prohlížeč v HTTP hlavičce cookie obsahující sessionID zpět na server, kde se porovná s hodnotou tam uloženou. Pokud hodnoty souhlasí, je totožnost uživatele ověřena.

Předávání session ID v odkazech se používá pokud jsou zakázány cookies v prohlížeči uživatele. Po nalogování do webové aplikace obsahuje každý odkaz v parametrické části URL unikátní číslo SessionID. URL odkazů mají tvar např.:

```
<a href=nejaka_stranka.php?SessionID=1234567890ABCDEF>
```

a do všech formulářů je vloženo skryté pole :

```
<input type="hidden" name="SessionID" value="1234567890ABCDEF">
```

Takto se unikátní identifikátor SessionID předává ze stránky na stránku. Při ztrátě "kontinuity" prohlížení je nutné se přelogovat. Tato metoda se často používá i dnes, nicméně z programátorského hlediska představuje jistou zbytečnou námahu navíc.

Různé implementace bezpečného logování mohou v URL (vlastně vůbec ve všech třech případech) přenášet libovolný řetězec, nemusí vždy jít o jakési IDčko "sezení". Ať už jde o bezpečnostní kód nebo cokoliv jiného, "dobyť" této hodnoty obvykle znamená možnost zcizení identity (jednání pod cizím jménem apod...), tedy prolomení bezpečnosti systému a průnik dovnitř.

### 7.3.3 Zcizení session

Pokud není spojení klient-server zabezpečené nějakým asymetrickým šifrováním, je vždy možné z odposlechu komunikaci okopírovat a vydávat se tak za někoho jiného (za nalogovaného člověka). Tomuto termínu se odborně říká "zcizit session".

Klíčem k prolomení bezpečnosti webového serveru je získání sessionID. Jakmile útočník sessionID získá, může přistupovat k neveřejným datům uživatele a získat neoprávněnou kontrolu nad systémem.

## 7.4 SQL databáze

### 7.4.1 Vkládání cizích SQL podmínek (SQL injection)

Jeden okruh bezpečnostních problémů představuje ukládání uživatelem zadaných dat do SQL databáze (např. MySQL). Ve spojitosti se speciálním významem, který mají

znaky uvozovky a apostrofy v SQL dotazech je možno významně pozměnit efekt SQL dotazů spouštěných serverovými skripty. Tyto speciální znaky je potřeba v uživatelském vstupu nahradit příslušnými HTML-entitami, aby nemohly měnit SQL dotazy.

Dobrou manipulací se speciálními znaky v uživatelském vstupu lze např. dosáhnout změny hesla ve webové aplikaci. Dotaz, který je umístěn v serverovém skriptu zpracovávajícím odeslaný formulář a měl by heslo měnit, může vypadat například následovně (PHP) : `UPDATE accounts SET password = '$newpass' WHERE id = $id.` Uživatelské ID je sice skryté ve webové stránce, ale ve zdrojovém kódu je snadno k nalezení a např. pomocí Javascriptu aplikovaného do adresního řádku prohlížeče lze hodnotu ID pozměnit tak, aby došlo k modifikaci SQL dotazu s cílem vlámat se do systému. Představme si, že vychytralý hacker pozmění hodnotu neviditelného, ale přepisovatelného `userID` třeba tak, že nahradí hodnotu `123456` hodnotou `123456 OR id=id.` SQL příkaz se pak trochu pozmění, ale s katastrofálními důsledky takto: `UPDATE accounts SET password='heslo' WHERE id=123 OR id=id.` Původní číselnou hodnotu `123456` jsme tedy změnili na „`123456 OR id=id`“, která způsobí, že složený výrok (nebo, or) bude pravdivý vždy a libovolné hackerem zadané heslo bude vyhodnoceno jako správné! Je to docela šílené...

Obrana spočívá v analýze a úpravě všech uživatelských vstupů do databáze a odstranění, nebo pozměnění nebezpečných znaků. Server-side skriptovací jazyky (PHP, ASP...) dnes již mají zabudovány funkce, které nebezpečné znaky vhodným způsobem zakódují. Na úrovni HTML se znaky `<` `>` a `"` převádějí na `&lt;` `&gt;` a `&quot;` (v tomto pořadí), na úrovni SQL se používá zpětného lomítka před nebezpečným znakem, tedy třeba `\'` se pak nepovažuje za „funkční“ apostrof. Záleží na konkrétním jazyku, v PHP se například používají funkce `addslashes()` a `stripslashes()`. Poznámka: i když v SQL dotazu porovnáváme sloupec s číselnou hodnotou, často je možné psát ho do apostrofů (třeba u MySQL), zde ukazovaný chybný příklad by měl správně končit `MySQL_Query(,... WHERE id=““.AddSlashes($pass).“...“).`

## 8 VÝHODY WEBOVÝCH TECHNOLOGIÍ

Dnešní webové technologie nabízí kreativnější a interaktivnější metody, pomocí kterých se mohou středně velké firmy zaměřit na nové cílové skupiny a měřit marketingové výsledky. Před zapojením do tohoto nového rychle se rozvíjejícího světa marketingu je však třeba pečlivě prověřit možnosti a potenciální problémy.

### 8.1 Souhrnně:

- na straně klienta stačí webový prohlížeč
- jednoduchá údržba – změny pouze na straně serveru
- aktuálnost – každá úprava se okamžitě projeví
- nižší nároky na HW klientů – stačí PC s webovým prohlížečem
- nižší provozní náklady
- nezávislost na platformě (OS) na straně klienta
- výborná dostupnost, možnost využití v lokální síti (Intranetu) i v Internetu
- v porovnání s tradičními metodami je online marketing stále flexibilnější a cenově dostupný pro malé a středně velké společnosti
- propagace v sociální síti a zejména blogy vyžadují pravidelnou aktualizaci obsahu a konceptu
- softwarové služby, které sledují chování zákazníků na webu, zjednodušují měření výsledků

#### 8.1.1 Weby sociálních sítí

Weby, jako například YouTube, MySpace a Facebook v USA, Friends Reunited ve Velké Británii, Cyworld v Jižní Koreji, Mixi v Japonsku a další, umožňují lidem odesílat obsah, například videa nebo osobní profily. Tyto weby jsou velmi oblíbené u mladých lidí, proto společnosti začínají investovat do reklamy v sociálních sítích. (Společnost Microsoft poskytuje technologii digitální reklamy pro server Facebook.)

**Výhody:** Marketing na těchto webech pomáhá společnostem oslovit mladší cílové skupiny, které začínají ignorovat tradiční reklamu. Například společnost Unilever



propaguje svůj deodorant Axe na stránce webu MySpace, která je určena pro tzv. Gamekillers — lidi, kteří se míchají do komunikace mladých snažících se najít lásku.

### 8.1.2 Blogy

Blogy, online deníky s komentáři a konverzací, najdete všude na Internetu. V podnikatelské sféře se nyní používají k propagaci produktů nebo k vyvinutí obrazu značky. Například inženýři společnosti Microsoft vytvářejí blogy o aplikaci Windows Internet Explorer a ostatních produktech, zatímco společnost General Motors provozuje blog zabývající se tématy od automobilových závodů po design automobilů.

**Výhody:** Při správném provozu poskytují blogy propagátorům možnost navázat neformální dialog se zákazníky. Společnosti mohou například testovat nové návrhy produktů, aby zjistily reakce zákazníků. Blogy jsou většinou nenákladné, jejich návrh stojí od 2 do 5 tisíc USD.

### 8.1.3 Podcasting

Podcasty jsou zvukové programy, které si lidé mohou stáhnout a kdykoli přehrát v přehrávači MP3. Tato vysílání mohou někoho zaujmout stejně jako dobrá kniha. Například americký výrobce domácích přístrojů Whirlpool vytvořil seriál 20-ti minutových, rodinně orientovaných vysílání zaměřujících se na různá témata, jako je tvorba dětských kostýmů nebo zbavení se závislosti na hazardních hrách.

**Výhody:** Ačkoli tato vysílání přímo neprodávají produkty společnosti Whirlpool, snaží se spojit zákazníky se značkou Whirlpool dříve, než koupí myčku nádobí nebo sušičku, vysvětluje Audrey Reed-Granger, ředitelka oddělení public relations společnosti Whirlpool, která tato vysílání vytvořila. „Jedná se o lidi, kteří chtějí poslouchat dobré programy při jízdě autem nebo jako kulisu při práci doma,“ říká.

### 8.1.4 Inteligentní tiskové zprávy

Tyto přímé tiskové zprávy pro zákazníky týkající se produktů nebo služeb společnosti obsahují často vyhledávané termíny. Společnosti odešlou tyto zprávy na zpravodajské weby online, které je indexují nebo „agregují“ a odešlou titulky předplatitelům.

**Výhody:** Například společnost prodávající marketingové nástroje či služby může napsat zprávy obsahující termíny jako „zrychlení prodejního cyklu“, které nasměrují lidi k informacím o společnosti. Aplikace Windows Internet Explorer 7.0 bude nabízet příležitosti v této oblasti, protože obsahuje funkce zjednodušující přihlášení k odběru zpravodajských kanálů.

### 8.1.5 Cílená propagace

Reklama ve vyhledávacích nástrojích, kdy společnosti platí za zobrazení svého webu ve výsledcích vyhledávání, je oblíbená již několik let. Nyní však mohou společnosti zaměřit svou reklamu v rámci kontextu, který lidé čtou online.

**Výhody:** Pomocí technologií a služeb společností, jako jsou AdValiant, Touch Clarity a Pulse 360, se například reklama společnosti poskytující hypotéky může zobrazit v článcích o rekonstrukcích domů nebo o nemovitostech. Mohou také zaměřit reklamu na čtenáře v geografické oblasti, kde hypoteční společnost působí.

## 9 NEVÝHODY WEBOVÝCH TECHNOLOGIÍ

### 9.1 Souhrnně:

- nehodí se pro některé typy aplikací
- vysoká závislost na poskytovateli aplikace
- nemožnost práce v offline režimu (začíná se řešit)
- omezené možnosti uživatelského rozhraní
- omezené možnosti validace dat na straně klienta (webový prohlížeč)
- nedokonalá podpora standardů (HTML, CSS, JavaScript) v prohlížečích
- bez-stavová komunikace při použití protokolu HTTP/HTTPS (lze obejít)
- množství přenášených dat (značkovací jazyk)
- problémy s bezpečností (webový prohlížeč, dostupnost v Internetu)

#### 9.1.1 Statické webové stránky

Představují soubor jednotlivých a pomocí odkazů vzájemně propojených webových stránek (HTML stránky). Každá stránka je "hotová" od A do Z a ve stavu, v jakém ji zanechal její autor. Nevýhodou těchto stránek je jejich obtížná udržitelnost při nárocích na další rozšiřování webových stránek a zejména fakt, že nedokáží zobrazovat data uložená v databázi. Další a podstatnou nevýhodou je, že stránky může tvořit pouze osoba znalá alespoň základů HTML a internetových technologií. Třetí nevýhodou je nemožnost zapojení jakékoliv přídavné inteligence do statických www stránek. Na fulltextové vyhledávání ap. tedy můžete zapomenout.

#### 9.1.2 Dynamické www stránky

Základem je programovací technologie, program (software) a data uložená v databázi. Program (online obchod, redakční systém, elektronická aukce ap.) umístěný na internetovém serveru. Pak na základě požadavku na zobrazení konkrétní stránky webovou stránku dynamicky sestavuje a odesílá do prohlížeče uživatele. Zjednodušeně se dá říci, že stránka je sestavena na základě "šablony" určující vzhled a dat získaných

z databáze (data jsou například seznam zboží, článků nebo konkrétní článek). Sestavení zajišťuje program běžící například na technologii ASP nebo ASP.NET. Dynamické stránky tedy fyzicky neexistují v uloženém stavu, na rozdíl od stránek statických.

## 10 NOVÉ TRENDY VE VÝVOJI WEBOVÝCH PROHLÍŽEČŮ

Společnost Sun hraje hlavní roli ve všech fázích vývoje webu, od serverů přes programovací jazyky až po standard Web 2.0.

„Sun zaujímá na webu klíčové postavení, stejně jako technologie Java a operační systém Solaris OS. Web však znamená mnohem více než jen tato jména, míní ředitel společnosti Sun pro webové technologie a spoluautor jazyka XML.

Vše, čím může Sun přispět k jeho lepšímu fungování, rychlejšímu růstu, snadnější použitelnosti a přístupnosti pro větší počet lidí, je dobré pro web jako takový, pro svět i pro samotnou společnost Sun, dodává Bray. Patří sem dodržování standardizačních pravidel, zachovávání otevřenosti a snaha být dobrými občany sítě. A přesně to je jádrem vize naší společnosti.

Společnost Sun hraje aktivní roli ve vývoji webu díky autorství dvou programovacích jazyků - dnes všudypřítomného jazyku Java a jazyku XML (Extensible Markup Language).

Inženýři společnosti Sun neusnuli na vavřínech a pracují na webu nové generace. Dnes se podílí na práci skupiny, která se v rámci sdružení IETF (Internet Engineering Task Force) zabývá projektem TRILL (Transparent Interconnection of Lots of Links). Jsou přesvědčeni, že TRILL by mohl nahradit technologii bridgingu a přispět k lepší stabilitě, menší zranitelnosti a kvalitnějšímu směrování v sítích.

Společnost také pracuje na zdokonalených směrovačích, které přinášejí lepší ochranu proti útokům typu DoS (Denial-of-Service) a na technologii "zaručeného odstranění" (assured delete), která umožňuje, aby soubory v šifrovaných systémech po určité době mohly zaniknout bez možnosti obnovy. Podobně se podílí na vývoji technologií nové generace v rámci pracovní skupiny IETF. Pod názvem Atom se skrývá standardizovaný protokol pro úpravy webových zdrojů, jako jsou blogy a deníky online. Sun se nejvíce zajímá o současné trendy společností typu Flickr a del.icio.us, které nabízejí software umožňující lidem spoluvytvářet obsah Internetu, nikoli jej jen pasivně konzumovat.

## 10.1 Web 2.0

Mnozí říkají, že web vstoupil do další etapy svého vývoje. Nová generace služeb online založená na sdílení a spolupráci v širokém měřítku bývá označována jako Web 2.0. Tim O'Reilly, prezident a výkonný ředitel společnosti O'Reilly Media, Inc., charakterizuje Web 2.0 jako "systémy, které jsou tím inteligentnější, čím více lidí je používá."

Společnost Sun se na vývoji platformy Web 2.0 podílí stejně intenzivně, jako tomu bylo u první generace sítě. Technologie Java nadále hraje hlavní roli ve vývoji pro web, stejně jako nové jazyky typu AJAX (Asynchronous JavaScript and XML). Řešení společnosti Sun společně pomáhají vývojářům obohacovat webové aplikace a dosahovat u nich podobné interaktivity jako u aplikací pro běžné počítače.

Společnost sice s jistotou neví, jaká bude další významná aplikace pro web, je však přesvědčena, že půjde o mobilní web. "Vývoj jednou dospěje do bodu, kdy mobilní zařízení budou tvořit základ pro přístup k webu." Pracuje také s nadšením na technologii PKI (Public Key Infrastructure). "Aby věci mohly správně fungovat, je naprosto nezbytné, aby jedna strana mohla bezpečně zjistit veřejný klíč druhé strany. Proto se mimo jiné zabývají způsoby, jak zajistit, aby to nebylo nákladné. Kdybychom umožnili, aby zde jedna firma získala monopol, přišlo by nás to velmi draho." Tato snaha o otevřenost je pro vizi společnosti Sun příznačná. Chceme zajistit, aby připojení bylo dostupné pro všechny, bez ohledu na to, kde se nacházejí a jakou technologii používají. Patnáct let od doby, kdy web začal spojovat svět dosud nevídanými způsoby, je heslo "The Network Is The Computer" pravdivější než kdy dříve.

## 10.2 Webové inženýrství

Pojmem inženýrství se obecně rozumí systematická aplikace vědeckých znalostí při návrhu a tvorbě cenově efektivních řešení praktických problémů. Jestliže použijeme tuto obecnou definici inženýrství i na oblast webu, pak můžeme definovat pojem webového inženýrství jako aplikaci systematického, disciplinovaného, kvalifikovaného a kvantifikovatelného přístupu k vývoji, provozu a údržbě webových aplikací.

Pro počítačové odborníky se web zpočátku jevil jako „jiná aplikace distribuovaného prostředí“. Nyní se zdá, že je všeobecný souhlas s tím, že web je

především základní medium pro rozvíjení a vývoj aplikací. Aplikace, která dnes není přístupná z webu, jakoby neexistovala.

Až dosud se vývojáři aplikací sdružovali ve speciálních softwarových týmech a jen zřídka kombinovali prostředky z různých zdrojů, což dokonce byla často strategie i předních producentů software. Web však může propojovat velice různorodé servery, organizace a zdroje. Podobně, jako jsou dnes distribuovány zdroje informací, jsou distribuovány i problémy a jejich řešení – aplikace.

Existuje celá řada populárních metodologií pro návrh a implementaci aplikací. Specifické problémy identifikované při vývoji systému vede k úsilí v nalezení metodiky pro formální specifikaci systému. Cílem je v maximální míře specifikovat navrhovaný systém v efektivní diskusi nejen se zadavatelem systému, ale i s potenciálními koncovými uživateli hned na počátku práce na projektu. Webové aplikace zásadně komunikují se zcela neidentifikovatelnými uživateli a jejich požadavky.

Vývoj internetových služeb prodělal zejména v poslední době zásadní vývoj. Doba BBS (Bulletin Board systém), nepřehledných Telnetových terminálů, FidoNetu a potažmo i statických HTML stránek je dávno za námi. Dnešku vévodí zejména multimedia a interaktivní služby v podobě webových aplikací - webová aplikace je aplikace, která je přístupná pomocí webového prohlížeče.

Prohlížeč v tomto případě vystupuje v roli tenkého klienta (řekněme klient, který si nikde neukládá data ani stavy – ty jsou uloženy výhradně na straně serveru). Doslova raketový nástup služeb typu Flickr, Last.Fm, YouTube a aplikací společnosti Google spolu s bezpečným internetovým bankovníctvím a online nákupy žádají nové technologie a metody přístupu k tvorbě takovýchto aplikací. Z počátku se jednalo o tvorbu dynamicky generovaných stránek pomocí některého z mnoha skriptovacích jazyků (především Personal Home Pages či Active Server Pages) umožňujících připojení na databázi s přemírou funkcí napsaných v Java Scriptu a spouštěných v prohlížeči na straně uživatele, později Adobe Flash (Macromedia Flash, původně FutureSplash Animator) s vlastním jazykem – Action Script, Javové applety běžící v prohlížeči a další více či méně známá řešení různých tvůrců.

Zejména v poslední době nastala potřeba shrnout používané technologie do jednoho celku a tento celek nějak vhodně označit, proto vzniklo označení Web 2.0 (nutno

podotknout, že tato nálepka vznikla především z marketingových důvodů – technologiemi nabitý web potřebuje nějaký název, který bude znít moderně, aby přitáhl pozornost potenciálních zákazníků).

### 10.3 Vývoj softwareových vs. webových systémů

Vývoj webových aplikací má řadu zvláštností a specifíků, které ho odlišují od vývoje klasického softwaru, informačních systémů a aplikací. Pro většinu webových aplikací jsou charakteristické následující vlastnosti:

- webové aplikace jsou v neustálém vývoji
- v mnoha případech není na počátku svého vývoje, proto není snadné odhadnout, co bude systém obsahovat, neboť jeho složení se bude postupně vyvíjet – a to jak během vývoje, tak především při jeho používání
- webové aplikace, na rozdíl od běžných systémů, jsou neustále ve fázi vývoje, mění se požadavky na ně kladené a roste jejich využití
- webové aplikace musí být navrženy s ohledem na velké množství uživatelů, které je budou využívat
- tito uživatelé, kteří mohou v prostředí internetu přistupovat prakticky z libovolné části světa, mohou (a jistě budou) mít rozdílné nároky, na aplikace samotné
- datová struktura webových systémů je do značné míry ovlivněna možnostmi, které mají nabízet
- vývoj samotných systémů proto musí vzít v úvahu jejich informační hodnotu, jejich vytváření, údržbu a začleňování do aplikací
- od webových systémů se obecně očekává, že budou po vizuální stránce maximálně propracované, budou využívat multimediální data a nabízet příjemné a intuitivní prostředí a ovládání
- webové aplikace musí většinou být vytvářeny za kratší dobu, než je tomu v případě klasických informačních systémů
- u webových aplikací klademe důraz na bezpečnost a ochranu osobních údajů
- webové aplikace jsou snadno a rychle přístupné uživatelům internetu



- stejně snadno a rychle se však projeví i nespokojenost a ztráta klientů, když systém nesplňuje jejich požadavky či dokonce chybí
- vývoj nových webových systémů vyžaduje spojení těch nejnovějších avšak často rozdílných technologií, standardů a komponent (skriptovací jazyky, HTML, databáze, multimédia, uživatelská rozhraní apod.)

Výše zmíněné charakteristiky webu a aplikací v něm pracujících nás jen utvrzují v tom, že vývoj těchto systémů je opravdu diametrálně odlišný od vývoje klasických aplikací.

## 11 NOVÉ TRENDY VE VÝVOJI WEBDESIGNU

Website (websajt, webové sídlo) je kolekce informací o určitém tématu nebo subjektu. Budovat website znamená navrhnout a vytvořit webové stránky, které naopak společně tvoří website. Webová stránka je tvořena informacemi, pro které je website vyvíjen. Z jiného pohledu je webová stránka jako stránka v knize a website je jako kniha samotná.

Samotná webová stránka se skládá z textu a z obrázků. Jinými slovy, webová stránka je soubor, který obsahuje všechna potřebná data. K podpoře dat nebo informací jsou ke stránce připojeny obrázky. První webové stránce webového sídla se říká Home Page (domovská stránka). Každá jednotlivá webová stránka, kterou website obsahuje, je HTML soubor a každá má svou vlastní webovou adresu. Webdesign vyžaduje určité aspekty a prvky, které je nutné zohlednit, má – li se webová stránka dostat k uživateli.

Prvním krokem je optimalizace stránky. Optimalizace stránek znamená zvýšení její viditelnosti pro vyhledávače (např. Google, Seznam apod.). Stránky by měly být, pokud možno, na prvních třech stránkách výsledků vyhledávání na dané klíčové slovo. Taková pozice znamená nárůst počtu návštěvníků. Obsah by měl být přehledně uspořádaný, informativní a lehce srozumitelný pro každého návštěvníka.

V neposlední řadě by měly webové stránky jasně ukazovat účel a záměr, pro který byly vytvořeny. Profesionální nebo obchodní webové stránky by neměly ničím mást nebo rušit a návštěvník by měl být schopen snadno se při hledání informací po webu pohybovat.

Protože mnoho webových vývojářů je zaměřeno především na grafické umění, věnují často příliš mnoho pozornosti tomu, jak webová stránka vypadá bez ohledu na to, jak návštěvníci stránku najdou. Na druhé straně odborníci na optimalizaci pro vyhledávače, SEO konzultanti, jsou posedlí tím, jak web funguje: jak velký provoz je generován a jaký je konverzní poměr. SEO vyžaduje mnoho nepěkných věcí jako např. dlouhé a nečitelné texty plné klíčových slov, odrážkové seznamy a mnoho textových odkazů. Grafici zase často stránky zahlťují množstvím grafických prvků, které iritují návštěvníky dlouhou dobou načítání stránky a ztěžují nalezení stránky vyhledávačem. Kvalitní webové stránky jsou dobrým kompromisem mezi oběma úhly pohledu na webdesign.

## 11.1 Tři pilíře moderního webu

Webová tvorba se sice ubírá mnoha směry, ovlivňuje ji několik různých názorových škol a plní ostatně i různé účely, přesto však v posledních pár letech na povrch zřetelně vyčnívají tři dominantní trendy, které si dovolím přirovnat ke třem pilířům moderního webu:

- důraz na uživatelský prožitek
- důraz na přístupnost
- důraz na ekonomickou návratnost a efektivitu

## 11.2 Důraz na uživatelský prožitek

Teoretické zkoumání působení webu na uživatele i praktické experimenty posledních několika let vyústily v koncepci uživatelského prožitku (user experience) jako hlavního cíle snažení webových tvůrců. Koncept uživatelského prožitku má již dnes dostatečnou teoretickou základnu a dále se dynamicky rozvíjí. Rychle přibývá nová literatura i konkrétní aplikace na něm založených metodik.

Ačkoli je pojetí uživatelského prožitku relativně nové a specifické pro web, ve skutečnosti staví na několika starších disciplínách pocházejících ještě z předwebové éry a původně dokonce ani nesouvisejících s informačními technologiemi. Jedná se o informační architekturu a použitelnost (usability).

Informační architektura pochází pravděpodobně z knihovnictví, ale její výsledky najdeme např. i v informačních systémech veřejných budov (cedulky na nádraží). Předmětem zkoumání teoretické, respektive cílem aplikované informační architektury, je hledání takových uspořádání a forem informací, které umožňují jejich nejefektivnější využití.

Na webu spolu přichází do styku značný objem informací s nemalým počtem uživatelů, z nichž každý má trochu jiné potřeby, požadavky, vzdělání a kulturní zázemí. Nelze si představit, že by jejich interakce mohla být optimálně řešena jen intuicí webových tvůrců, bez velmi solidních teoretických základů. Je proto zřejmé, že web a informační architektura se dříve či později musely potkat, a že se jedná o svazek velmi perspektivní.

Použitelnost je také oborem, který se uplatňoval již dříve, např. při konstrukci ovládacích prvků přístrojů a zejména při návrhu uživatelského rozhraní počítačových programů. I zde platí, že intuice a zkušenosti návrháře k dosažení perfektního výsledku nestačí. Dynamicky se proto rozvíjejí exaktní metody projektování a testování použitelného rozhraní.

Webová stránka, a to i v případech, kdy má převážně zábavní funkci, je vždy nástrojem, kterým uživatel dosahuje svého cíle. Použitelnost stránky pak přímo podmiňuje úspěšnost, s jakou uživatel svého cíle dosáhne a tím i kvalitu jeho uživatelského prožitku. Aplikace vědecky založené použitelnosti na web je proto naprosto logická.

### **11.3 Důraz na přístupnost**

Druhým výrazným trendem v oblasti tvorby webových stránek je důraz na jejich přístupnost (accessibility). Ta má, trochu zjednodušeně řečeno, dvě polohy: správnou funkčnost stránek v různých zařízeních na různých platformách a bezbariérový přístup pro osoby s různou povahou i mírou zdravotního postižení.

Multiplatformnost webu jistě není ničím novým. Právě naopak, web byl jako multiplatformní stvořen a má tuto charakteristiku geneticky zakódovanou. Zatímco však dříve multiplatformnost znamenala především možnost přístupu k webovým dokumentům různými prohlížeči z různých počítačů a operačních systémů, v poslední době nabývá nového rozměru. Uživatelé začínají k webovým stránkám přistupovat pomocí zařízení, která bychom dříve za počítač neoznačili. Dnes to jsou mobilní telefony, co to bude zítra, můžeme jen hádat.

Snad ještě významněji se trend přístupnosti projevuje ve vztahu ke zdravotně znevýhodněným uživatelům. Moderní společnost klade značný důraz na podporu občanů se specifickými potřebami a aktivně brání jejich diskriminaci. Ve většině států zákony předepisují bezbariérovou konstrukci veřejných budov a komunikací, někde a za určitých podmínek pak i provozoven soukromých firem.

### **11.4 Důraz na ekonomickou návratnost a efektivitu**

Poslední trend, který chci dnes zmínit, souvisí s ekonomickými aspekty webové tvorby. Ještě nedávno jsme byli svědky dvou extrémů. Někdo vkládal do webu bez většího

rozmyslu značné finanční částky, jiný do něj naopak neinvestoval vůbec. Ač se jedná o přesně opačné přístupy, jednu věc mají společnou - nízké vědomí toho, jaké efekty je web schopen přinést a kolik se tedy vyplatí investovat, aby se vložené prostředky v rozumné době vrátily.

Dnes je situace jiná a návratnost investic (ROI) se skloňuje ve všech pádech. Není však jen pouhým přáním, ale přesouvá se, podobně jako jiné aspekty webové tvorby, do roviny vysoce odborného zkoumání a exaktního plánování. Ve skutečnosti oba předchozí pilíře, uživatelský prožitek, i přístupnost, mají též svůj ekonomický rozměr, pokud dokonce nejsou požadavky na návratnost investic přímo motivovány.

Na druhou stranu si ale musíme připustit, že ekonomika webu je zatím zřejmě nejméně poznanou a rozvinutou oblastí. Její skutečný rozvoj nás teprve čeká. Těžiště tohoto rozvoje však zřejmě nebude ležet na poli samotné tvorby webových stránek, ale spíše v oblastech internetového marketingu a obchodních strategií.

## 11.5 Budoucnost webových technologií

Až dosud jsem se v článku vyhýbal webovým technologiím, což mnohé z vás možná překvapilo. Vždyť právě technologie jsou nejčastějším předmětem prognóz a spekulací o tom, kam se bude web i celý Internet v budoucnosti ubírat. To, že jsem odsunul technologie až na závěr článku, však má své důvody:

- prvním důvodem je, že neumím předpovědět náhodný vznik nových technologií a porušil bych tedy svůj slib, že nebudu věštit z křišťálové koule
- druhým a podstatnějším důvodem je, že technologii přikládám spíše okrajový význam

Nové technologie sice mohou vznikat živelně a náhodně, širokého přijetí se jim však dostává jen podle toho, jak naplňují společenskou poptávku. Takovou poptávku dle mého názoru reprezentují tři pilíře moderní webové tvorby nastíněné výše - uživatelský prožitek, přístupnost a ekonomická efektivita. Je proto pravděpodobné, že právě ty budou diktovat směr technologického rozvoje webu v nejbližších letech.

Pro argumenty lze opět sáhnout ke dalším trendům, které se zřetelně projevují již nyní:

- komplexnost projektování uživatelského prožitku vede ke zvyšování počtu členů projektových týmů
- rozvoj automatizovaných systémů pro řízení webových projektů a správa obsahu
- hranice mezi projekty dělanými jen tak na koleni a těmi, které využívají pokročilejších autorských a správních systémů, se bude nepochybně přibližovat k čím dál menším projektům
- důraz na přístupnost stránek a zároveň ekonomičnost jejich tvorby již dnes vede k přijetí moderních metod tvorby založených na maximálně jednoduchém, standardizovaném a modulárním kódu stránek (striktní HTML, CSS, DOM)

## ZÁVĚR

Ačkoli to možná z perspektivy samotných webových tvůrců nemusí být jednoznačně patrné, web je více než technologií ovlivňován svou reálnou společenskou upotřebitelností a ekonomicko-podnikatelskými mantinely. Zatímco technologie již dosáhla relativně vysokého a pravděpodobně dostatečného stupně zralosti, ostatní aspekty webové tvorby zatím spíše vstupují do puberty a jejich dospívání budeme teprve svědky.

Osobně proto nevidím budoucnost webové tvorby ve výraznějších technologických změnách, nýbrž v profesionalizaci a odborném prohlubování činností směřujících ke zkvalitňování uživatelského prožitku a k maximalizaci přínosů webu pro podnikatelské subjekty. To vše ovšem v mezích racionálního ekonomického uvažování a s civilizovaným citem k potřebám handikepovaných uživatelů.

Jak tedy budou vypadat informační technologie budoucnosti a jak ovlivní člověka? Všudypřítomné technologie zřejmě postupně vyplní kybernetický prostor. Člověk se naučí žít s různými implantáty rozšiřujícími jeho schopnosti včetně přístupu na Internet. Rychlost vývoje hardwaru zřejmě nebude stále exponenciální, neboť bude limitována nutností manipulací na atomární a subatomární úrovni a zaostáváním vývoje softwaru.

Odborníci na virtuální realitu předpokládají, že se nůžky mezi kvalitou hardwaru a nedokonalostí softwaru, který již dnes dosáhl mezní komplexity, budou nadále rozšiřovat. Za 50 let možná zabere údržba informačních systémů značnou část provozního času. Jedinou možností, jak se tomuto scénáři vyhnout, je přijít s nějakou zásadně odlišnou koncepcí vývoje programů.

Podle počítačových inženýrů na Michiganské univerzitě běžná zařízení (telefon, kamera, poznámkový blok) splynou do jediného a všude kolem nás se budou pohybovat specializovaní i relativně univerzální ro-boti.

Technologie bude s největší pravděpodobností velice mobilní a tak malá, že bude prakticky neviditelná. Každý bude on-line. Obraz si necháme promítat přímo na sítnici. Výpočetní substrát bude všudypřítomný. Informačně-technologické oddělení se bude zabývat bezpečností a zejména ochranou proti softwarovým patogenům.

Vědci z Harvardu uvádí, že za 50 let se prostřednictvím virtuálního prostředí bude realizovat i terapie. Implantáty budou automaticky monitorovat nejen tělesné funkce, ale i duševní stavy člověka. Počítače nošené na těle se podle nich přizpůsobí svým

nositelům, stanou se skutečně „osobními“. Tato technologie donutí člověka redefinovat i pojem soukromí.

Podle odborníků na umělou inteligenci veškeré encyklopedické informace budou za 50 let snadno dostupné všude kolem nás. Inteligence přestane být schopností získávat odpovědi na otázky. Odborníci srovnávají přicházející revoluci s objevem kalkulačky, která člověka zbavila nutnosti provádět ručně složité výpočty a umožnila mu soustředit se v matematice na meritum věci. Podle nich zaniknou klasické univerzity a vlády ztratí kontrolu nad vzdělávacím procesem.

Technologie bude podle současných profesorů informatiky za 50 let na jedné straně všudypřítomná, na druhé straně se na ni lidé nebudou fixovat a de facto je nebude ani příliš zajímat. Díky kybernetickému prostoru zmizí též potřeba se přemísťovat. To přitom nemusí znamenat konec automobilů, řídit se však bude výhradně pro zábavu. Zaniknou rovněž města – nebude žádný důvod, proč by se lidé měli soustřeďovat. Profesoři jsou skeptičtí k elektronickým knihám a domnívají se, že i za 50 let budou knihy papírové. Podle nich však zcela zaniknou kamenné obchody i kanceláře, on-line vzdělávání způsobí konec klasických univerzit. Udrží se pouze prestižní školy, jejichž diplomy půjde rychle transformovat na pracovní pozici a plat.

Mají také představu, že obyvatelé vyspělých zemí budou většinu svého času trávit ve virtuálních prostředích podobných dnešním on-line světům – avšak mnohem dokonalejších a komplexnějších. Preference určitých virtuálních prostředí pak bude i tím, co bude vytvářet pojítko mezi lidmi.

Na univerzitách se nyní připravují studenti, kteří již pravděpodobně budou v podobném světě žít a pracovat. Objem relevantních informací, s nimiž pracují, extrémně vzrůstá. Bez odpovídajících informačních technologií se nemohou obejít. Odpovídá náplň a způsob současného vysokoškolského studia potřebám budoucnosti? Jaké nás budoucnost potřebuje?



## ZÁVĚR V ANGLIČTINĚ

The web is affected more by the real usability of customers and economical and business barriers than by technologies. While technologies have reached relatively high and probably sufficient level of maturity, other aspects of web designing more likely achieve the period of adolescence and we will be the witnesses of their adulthood.

Personally, I don't see the future of web design in significant technologic changes but in innovation and skilled deepening of activities, which focus on improving quality of user's enjoyment and maximization of web gains for entrepreneurial subjects. At the same time must be respected needs of handicapped inhabitants and rational economical consideration.

What will be the information technologies like in the future and how will they affect people?

Omnipresent technologies will apparently gradually fill the cybernetic space. People will probably learn to live with different implants, which will extend their abilities including Internet access. The speed of hardware development will apparently not be still exponential, because it will be limited by necessity of manipulation and backwardness of software development.

Specialists in virtual reality suspect that the contrast between the quality of hardware and the imperfection of software (which today reached its own limits) will still be more significant. In fifty years the maintenance of information systems will perhaps fill most of the operating time. The only possibility how to avoid this scenario is to set up completely different conception of programme development.

According to computer experts from Michigan University common equipments such as telephone, video or notebook will be integrated into one and specialized or relatively universal robots will move around us.

Technologies will probably be very mobile and very small so nearly invisible. Everybody will be online. Pictures will be screened directly on the retinae. The department of information technologies will deal with the safety and protection against software pathogenic factors.

Scientists from Harvard claim that therapy will be realized through the virtual space. Implants will automatically monitor not only physical functions but even peoples' mental state. Computers worn on the body will assimilate with their wearers, they will become really personal computers. This technology will force people to re-pose the idea of privacy.

According to specialists in artificial intelligence, all information from encyclopedias will be easily available everywhere around us in fifty years. Intelligence will no more be the ability to get the answers to the questions.

Experts compare the coming revolution with the invention of calculator, which helps people to concentrate on facts in issue in mathematics and they do not have to solve difficult mathematic problems. According to them the classical universities and governments will loose the control under the educational process.

According to the current professors of computer science, technology will be everywhere in fifty years, on the other hand people will not be interested in it and will not fixate on it. The need of transfer will fade away because of cybernetic space. It doesn't have to mean the end of automobiles, but people will drive their cars just for fun. Towns will not exist as well – for people there will not be reason to gather.

Professors are sceptic to electronic books and they assume that even in fifty years there will still be paper books. According to them, the shops and offices will not exist, online education will cause the end of classical universities. There will be only prestigious schools, whose diplomas will be easily possible to transform to the salary and working position.

They also think that the inhabitants from industrial countries will spend most of their time in virtual spaces, which will be similar to our online worlds – but these spaces will be more perfect and complex. Preference of certain virtual spaces will make the link between people.

Students, who study now at universities, will probably live and work in a similar world. Capacity of relevant information, which they work with, has extremely increased. They cannot work without appropriate information technologies. Does the contents and form of the university studies agree with the needs of future? What should people be like in the future?

## SEZNAM POUŽITÉ LITERATURY

- [1] BELLINASO, Marco. *Webové programování v ASP.NET 2.0 Problém, návrh, řešení*. 1. vyd. Praha: Computer Press, 2007. ISBN 978-80-251-1893-1.
- [2] BEDNÁŘ, Vojtěch. *Alternativní webové prohlížeče Firefox, Opera, Mozilla*. 1. vyd. Praha: Computer Press 2006. ISBN 80-251-0566-0.
- [3] CEDERHOLM, Dan. *Webdesign s webovými standardy*. 1. vyd. Praha: Zoner Press 2004. ISBN 80-86815-15-3.
- [4] *Internet: vývoj Internetu* [online]. [cit. 2007-11-13]. Dostupný z WWW: [<http://www.wikipedia.org/>](http://www.wikipedia.org/).
- [5] *Internet: historie Internetu* [online]. [cit. 2008-03-10]. Dostupný z WWW: [<http://www.ics.muni.cz/>](http://www.ics.muni.cz/).
- [6] *Java: vývoj webových stránek* [online]. [cit. 2008-01-25]. Dostupný z WWW: [<http:// www.lupa.cz/>](http://www.lupa.cz/).
- [7] *Standardy: webové standardy* [online]. [cit. 2007-12-04]. Dostupný z WWW: [<http:// www.developer.mozilla.org />](http://www.developer.mozilla.org/).
- [8] *Statistiky: návštěvnost webu* [online]. [cit. 2008-02-07]. Dostupný z WWW: [<http:// www.navrcholu.cz />](http://www.navrcholu.cz/).
- [9] *Bezpečnost webu: webové stránky* [online]. [cit. 2008-01-20]. Dostupný z WWW: [<http:// www.toplist.cz />](http://www.toplist.cz/).
- [10] *Jazyk PHP: jak to funguje* [online]. [cit. 2008-04-15]. Dostupný z WWW: [<http:// www.php.cz />](http://www.php.cz/).
- [11] *ASP: výhody ASP* [online]. [cit. 2008-03-08]. Dostupný z WWW: [<http:// www.asp.net.cz />](http://www.asp.net.cz/).

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

HTML	HyperText Markup Language - značkovací jazyk pro hypertext.
WWW	World Wide Web - celosvětová síť s protokolem HTTP.
PDF	Portable Document Format – formát pro přenositelné dokumenty.
ARPA	Advanced Research Projects Agency - Pentagonská agentura.
UCLA	První uzel americké sítě.
NCP	Network Control Protocol - protokol pro komunikaci.
TCP/IP	Sada síťových protokolů.
Ethernet	Jeden z typů lokálních sítí.
SGML	Programovací jazyk, předchůdce HTML.
FDDI	Fiber distributed data interface - síť s kruhovou topologií.
NSF	National Science Foundation - Národní vědecká nadace USA.
EARN	European Academic and Research Network – Internetová síť.
FESNET	Federal Educational and Scientific NETwork - celorepubliková páteřní síť.
CESNET	Czech Educational and Scientific NETwork – česká páteřní síť.
TEN-34	Trans-European Network Interconnect at 34 Mbit/s – celoevropská síť.
ISP	Internet Service Provider - poskytovatel přístupu k Internetu.
LAN	Local Area Network - lokální síť, místní síť.
SGML	Standard Generalized Markup Language – značkovací jazyk.
DTD	Document Type Definition, - definice typu dokumentu.
CSS	Cascading Style Sheets - tabulky kaskádových stylů.
DHTML	Dynamické HTML stránky.
XML	Extensible Markup Language - rozšiřitelný značkovací jazyk.
XHTML	Extensible Hypertext Markup Language – rozšiřitelný značkovací jazyk.
FTP	File Transfer Protocol - protokol aplikační vrstvy z rodiny TCP/IP.

HTTP	Hypertext Transfer Protocol - internetový protokol.
HTTPS	Nadstavba počítačového protokolu HTTP.
PDA	Personal digital assistant - osobní digitální pomocník.
SGML	Standard Generalized Mark-up Language - metoda pro formování textu.
OS/2	Operační systém
IE	Internet Explorer – internetový prohlížeč.
NN	Netscape Navigátor – internetový prohlížeč.
RIA	Rich Internet Applications - webová aplikace pro surfování kyberprostorem.
C	Programovací jazyk.
C++	Objektově orientovaný programovací jazyk.
AJAX	Asynchronous JavaScript and XML – kombinace html/xml s JavaScriptem.
PHP	Hypertext Preprocesor – skriptovací programovací jazyk.
ASP	Active Server Pages - technologie nezávislá na programovacím jazyce.
Java	Objektově orientovaný programovací jazyk.
SQL	Structured Query Language - strukturovaný dotazovací jazyk.
URL	Uniform Resource Locator - jednotný lokátor zdrojů.
LINK	Hyperlink – odkaz.
DOM	Document Object Model – objektový model dokumentu.
HW	Hardware – technické vybavení počítače.
SW	Software - programové vybavení.
PC	Personal computer – osobní počítač.
MP3	Formát ztrátové komprese zvukových souborů.
ISDN	Integrated Services Digital Network - Digitální síť integrovaných služeb.
ADSL	Asymmetric Digital Subscriber Line - asymetrické připojení.
USB	Universal Serial Bus - univerzální sériová sběrnice.

---

Wi-Fi	Wireless fidelity – bezdrátová síť.
EDGE	Enhanced Data rates for GSM Evolution – vyšší stupeň v technologii GSM.
GPRS	General Packet Radio Service - mobilní datová služba.
CDMA	Code division multiple access - kódový multiplex CDMA.
PKI	Public Key Infrastructure - infrastruktura veřejných klíčů.
OOP	Object-oriented programming – objektově orientované programování
ASP	Active Server Pages - technologie nezávislá na programovacím jazyce
WML	Wireless Markup Language - značkový jazyk založený na jazyce XML
DLL	Dynymic Library – dynamická knihovna
PHP	Personal Home Page – skriptovací programovací jazyk

**SEZNAM OBRÁZKŮ**

Obr. 1. Mapa sítě CESNET v České republice.....	14
Obr. 2. Mapa sítě CESNET v zahraničí.....	15
Obr. 3. Aktuální zatížení pátečních linek CESNETu.....	16
Obr. 4. Implementace objektových postupů v PHP.....	35

**SEZNAM TABULEK**

Tab. 1. Přehled nejvýznamnějších poskytovatelů a jejich linky.....	18
Tab. 2. Hodnocení prohlížečů.....	45