

NOVINKY INTRANETU

Martin Molhanec

ČVUT-FEL, K-313, Technická 2, 166 27 PRAHA 6, Dejvice, ČR

Abstrakt

Příspěvek se zabývá novinkami v intranetových technologiích, které se objevily v roce 1999 a začátkem roku 2000. Článek si neklade za cíl podat úplný přehled všech novinek, ale pouze upozornit na ty nejdůležitější z nich podle názoru autora příspěvku.

Klíčová slova: Informační systémy, intranet, HTML, XHTML, Netscape, Gecko, Mozilla, Apache, XML, ASP, JSP, ZEND, IIS

Úvod

I v letošním roce 2000 si klade autor tohoto příspěvku za cíl seznámit účastníky semináře *Tvorba software* s novinkami v oblasti budování informačních systémů založených na technologiích intranetu. Protože tento příspěvek ve své podstatě navazuje na podobné příspěvky autora přednesené v letech 1999 a 1998, omlouvám se těm čtenářům, kteří si je doposud nemohli přečíst, nechci totiž v tomto příspěvku opakovat věci již jednou řečené, ale věnovat se těm věcem, které jsou nové a nebo nebyly zatím podrobněji zmíněny. Podobně je také výběr novinek podřízen vkusu autora, který se ovšem snaží být odborně nezaujatý, a jeho vědomostem v dané problematice a dostupnosti pramenů zejména na Internetu.

Pro účel klasifikace intranetových technologií jsem rozdělil svůj příspěvek na dvě základní části. Na část, která se zabývá pokrokem v technologiích na straně klienta, a na část, která se zabývá pokrokem technologií na straně serveru. Další stupeň dělení je na technologie implementačně nezávislé a na technologie implementačně závislé, které se dělí podle svých platform. Jako každé, není ani toto dělení naprosto dokonalé, protože některé technologie nelze snadno zařadit přesně do jedné kategorie.

1. Technologie na straně klienta

Vývoj technologií na straně klientů, čili prohlížečů je stále udáván soupeřením dvou firem, jejichž prohlížeče jsou v současné době nejrozšířenější. Jedná se pochopitelně o firmu Microsoft se svým prohlížečem *Microsoft Internet Explorer (MSIE)* a firmu Netscape se svým prohlížečem *Netscape Navigator*, respektive *Netscape Communicator*. Přestože má v současné době firma Microsoft lepší pozici, není situace jednoznačně rozhodnuta.

Vývoj technologií na straně klienta se ovšem netýká pouze samotného prohlížeče, jedná se zejména o další vývoj v oblasti HTML a dnes již i XML, a vývoj v oblasti DOM a skriptovacího jazyka na straně klienta.

1.1 Klienti

Pokrok na straně klientů (browserů, prohlížečů) je charakterizován neustálým bojem mezi firmou *Microsoft*, která někdy až silovým způsobem prosazuje svá řešení, někdy

nápaditá, někdy omezující, a stranou příznivců prohlížeče *Netscape*, který je v poslední době poněkud na pozadí zájmu, přestože se výsledky příznivců tohoto prohlížeče sdružených v týmu *Mozilla* mohou nakonec zhodnotit ve velmi zajímavé formě. Typickým znakem všech posledních verzí klientů je silná podpora nových standardů založených na XML.

1.1.1 Microsoft Internet Explorer (MSIE) 5.5

Poslední běžná verze MSIE je 5.01. Tato verze se však od starší verze liší zejména opravami chyb a nedostatků verze 5.0, zejména pro platformy Windows NT a Windows 2000 a přináší jenom některá drobná vylepšení samotného MSIE a také programu *Outlook Express*. Možná nejzajímavější novinkou je schopnost rozšířit program *MsInfo32.Exe* o schopnost poskytování informací o konfiguraci a verzi MSIE.

S druhou beta verzí produktu *Windows Millenium* se k potěše beta testerů objevila jako součást tohoto operačního systému, nástupce Windows 98 SE, i nová verze MSIE ve verzi 5.5. Nicméně se podle dostupných informací nejedná zatím o nijak revoluční produkt, i když přináší některé nové a užitečné funkce. Pro běžného uživatele je snad nejužitečnější funkce *preview* před tiskem HTML stránek. Je nutné snad zdůrazňovat, jak tato funkce doposud chyběla? Další nová funkce je zabudovaná *MSN Messenger Service*, která umožňuje interaktivní *chat* mezi uživateli MSN. Jako další vylepšení se uvádí robustní podpora *DHTML*, *CSS* a *SMIL (Synchronized Multimedia Integration Language)*.

Verze 5.5 MSIE je ke stáhnutí z webového sídla firmy Microsoft. Nicméně je podivuhodná skutečnost, že současná verze podporuje pouze Windows 9x a NT, nikoliv Windows 2k!?

1.1.2 Gecko

Gecko je název pro nový prohlížeč firmy Netscape. Starší název pro Gecko je *NGT (Netscape's next-generation browsing engine)* případně *Raptor* či *NGLayout engine*. Cílem projektu je vytvoření rychlého a malého klienta postaveného na otevřených standardech. Zajímavým rysem Gecka je skutečnost, že jeho jádro je postaveno na projektu *Mozilla*. Dochází zde k zajímavému procesu. Firma Netscape předala zdrojové kódy svého prohlížeče *Netscape Navigator* otevřenému projektu *Mozilla* a na výsledcích tohoto projektu staví svůj projekt *Gecko*! Samotný projekt *Gecko* má několik cílů:

- *Any Application*

Modulární struktura Gecka umožní jeho kód integrovat do libovolné aplikace

- *Any Operating System*

Browsovací engine Gecka bude snadno portovatelná na libovolný operační systém

- *Any Device*

Gecko je navrženo tak, aby mohlo fungovat nejen v osobním počítači, ale i v mobilních telefonech, PDAs, faxech, TV set-top boxech a dalších zařízeních.

- *Any Developer*

Softwaroví vývojáři budou mít otevřený přístup ke zdrojovým kódům Gecka. Kód bude zpřístupněn prostřednictvím projektu *Mozilla*.

Podobně jako jiné projekty i projekt *Gecko* staví na podpoře otevřených standardů:

- *HTML 4.0*

Plná podpora vyjma *bidirectional text layout*, který používá pouze hebrejščina a arabština. Nicméně na projektu této podpory zahájila práce firma IBM.

- *Style Sheets*
 - *CSS 1* - plná podpora
 - *CSS 2* - částečná podpora zahrnující *CSS2 positioning*
- *DOM*
 - *Level 0*
 - *Level 1 Core*
 - *Level 1 HTML*
 - *DOM 2*
Částečná podpora zahrnující *DOM 2 events* a *DOM 2 CSS interface*
- *XML 1.0*
- *RDF*
- *JavaScript 1.5*
Ve shodě s doporučením *ECMA-262 Edition 3 (ECMAScript)*
- Podpora protokolů *HTTP, FTP, Gopher*
- *SSL*
- *Unicode*
- *OJI (Open java Interface)*
- Podpora formátů obrázku
 - *PNG*
 - *GIF*
 - *JPEG, PJPEG*

1.1.3 Mozilla

Mozilla je otevřený projekt, jehož cílem je vytvoření standardního otevřeného a přenositelného prohlížeče. Projekt Mozilla vychází ze zdrojového kódu prohlížeče Netscape, který byl firmou Netscape projektu Mozilla předán při jeho zrodu. Projekt Mozilla je jako každý jiný otevřený projekt zcela mezinárodní a dobrovolný. Samotný projekt Mozilla se dělí na několik částí, vypíši zde pouze ty nejdůležitější, ostatně mnohé jeho další projekty jsou pouze interní, či se jedná o dílčí projekty ostatních projektů.

- *SeaMonkey*
Je projekt vlastního prohlížeče projektu Mozilla, který zahrnuje *NGLayout/Gecko layout engine* a je postaven na *XPCOM* komponentní architektuře.
- *Editor*
Je projekt HTML 4.0 kompatibilního WYSIWYG editoru. Tento editor bude též využit pro psaní zpráv v modulech *mail* a *news*.
- *Instant Messaging and Chat*
Je projekt podpory různých protokolů pro *messaging* a *chat*. Výsledkem tohoto projektu je například IRC client *ChatZilla*.
- *Open JVM Integration (OJI)*
Projekt zabývající se plug-in architekturou umožňující integrovat JVM do Mozilla browseru.
- *Blackwood*

Projekt zabývající se několika druhy integrace Javy a Mozilly. Jedná se o integraci s XPCOM, Java DOM API, OJI, Java WebClient API a Plug-iny implementovanými v jazyce Java (Pluglets).

- *JavaScript*
Projekt implementace jazyka JavaScript podle ECMA-262 (ECMAScript) specifikace.
- *Mail/News*
Projekt pro přístup k mailu a newsům, založený na technologiích XPFE, XPCOM a RDF. Projekt bude podporovat protokoly POP, SMTP, IMAP, NNTP a LDAP.
- *MathML*
Implementace W3C *Mathematical Markup Language*.
- *Netscape Portable Runtime (NSPR)*
Projekt na platformě nezávislého negrafického prostředí projektu Gecko

Celý projekt Mozilla je velice ambiciózní, ale to ostatně projekt *LINUX* taky. Nicméně skutečnost, že existuje šance na vytvoření modulárního, škálovatelného, standardního prohlížeče, který bude volně k dispozici všem vývojářům, je velice zajímavá a může ohrožovat dnes vedoucí pozici firmy Microsoft.

1.2 XHTML 1.0

Co je to *XHTML*? Jedná se o nový standard konsorcia W3C, který je odpovědí na otázku kudy dál s HTML a co udělat s XML. Zjednodušeně lze napsat rovnici: $XHTML = XML + HTML$. Trochu složitěji lze odpovědět, že XHTML je reformulace HTML 4.0 v XML 1.0. Co to znamená? XHTML je jazyk ekvivalentní jazyku HTML, dodržující však striktně pravidla XML. Jedná se tedy o čistě XML produkt. Toto pojetí má několik výhod. Jednak je jazyk XHTML snadno rozšiřitelný, což je umožněno samotným principem XML. Další výhodou by mělo být zvýšení portability a přenositelnosti XHTML aplikací, protože XML je dobrý základ pro uskutečnění tohoto cíle. Zjednodušeně řečeno, XHTML dává jednoznačnější výsledek při svém zobrazení.

Mezi HTML a XHTML existují rozdíly, nicméně již dnes je možné psát kód webových stránek v XHTML, protože XHTML je ve své podstatě ze shora kompatibilní s HTML. Opět lze zjednodušeně říci, že stránka napsaná v XHTML bude prohlížečem, který umí pouze HTML správně zobrazena, což ovšem neplatí naopak. Většina stránek HTML totiž bohužel nedodržuje ani správné zásady pro psaní HTML kódu, tak jak to vyplývá z definice HTML a tam, kde je definice HTML nejasná, tím spíše. Jaké jsou hlavní zásady pro psaní XHTML kódu?

- Všechny HTML značky musí být malými písmeny.
- Všechny atributy musí být v uvozovkách.
- Všechny neprázdné elementy musí být ukončeny.
- Elementy se mohou vnořovat, ale nesmí se překrývat.
- Vyžadované elementy nesmí chybět.
- Všechny dokumenty musí obsahovat *doctype* deklaraci.
- Kořenový element dokumentu musí být *html* element a musí specifikovat *xhtml* jmenný prostor.
- Prázdné elementy musí být ukončeny.
- Atributové páry nemohou být minimalizovány.

- *Script* a *style* elementy musí být označeny jako *CDATA* sekce.

Je patrné, že většina z výše uvedených zásad se dá poměrně snadno aplikovat již nyní, a mnohé z nich prostě odpovídají konvencím dobře psaného HTML kódu. Podstatná výhoda je však ta, že XHTML si tento styl prostě vynucuje. Pro usnadnění práce tvůrců stránek s HTML kódem již dnes existují konvertory z HTML do XHTML, které inteligentním způsobem původní kód opravují tak, aby byl správný podle XHTML specifikace. Jeden z nejlepších (*TidyHTML*) je dokonce zcela zdarma, takže není proč váhat. Vaše nové stránky mohou být pro nový standard již připraveny.

2. Technologie na straně serveru

Technologie na straně serveru jsou v současné době stále ve velice prudkém rozvoji. Ukazuje se totiž, že navrhování webových sídel, které v sobě na jedné straně obsahují skrytou obchodní logiku a na straně druhé je možné je označit za on-line publikační systémy, je nesmírně komplikované. První taková technologie *CGI* byla poměrně jednoduchá, umožnila při požadavku klienta o určitou stránku spustit libovolný program na serveru a jeho výstup vrátila klientu. Přestože byla vyvinuta velice komplexní řešení na základě této technologie, má tato technologie spoustu nedostatků. Snad nejvýraznějším z nich byla celková malá výkonnost uvedeného řešení. Při každém přístupu na stránku bylo nutné spustit aplikaci a tato činnost je ve většině operačních systémů poměrně časově náročná. Výhodou *CGI* řešení je poměrná nezávislost na platformě. Pokud byl program napsán ve vhodném, pro více platformem rozšířeném, programovacím jazyku.

Dalším krokem byla koncepce, která umožnila uživatelům rozšiřovat funkčnost webového sídla pomocí aplikací psaných přímo pro speciální rozhraní toho či onoho HTTP serveru. Výhodou této koncepce byla především velká výkonnost takto realizovaných aplikací. Nevýhodou pak omezení pouze pro platformu jednoho konkrétního serveru a současně nutnost psát aplikaci v jazyku C či C++. Logickým vyústěním těchto problémů pak je současná koncepce programovacích jazyků vyšší úrovně vnořených do HTML stránek. Vnoření těchto jazyků do HTML stránek umožnilo pro vývoj webové aplikace používat do určité míry WYSIWYG editorů a tím podstatně usnadnit vývoj. Nicméně celá koncepce vizuálního navrhování dynamických webových aplikací není stále uspokojivě vyřešena.

Když pomínu různá specializovaná nebo nepříliš často používaná řešení, lze nejrozšířenější současné serverové technologie rozdělit na technologie na platformě nezávislé, které jsou dnes realizovány zejména na platformě jazyků Java, JavaScript nebo Perl, a na řešení na platformě závislá, která jsou realizována pro určitý konkrétní HTTP server. Nejčastější řešení závislá na platformě jsou řešení pro *Internet Information Server (IIS)* firmy Microsoft, *Netscape Enterprise Server* a *Apache Server*.

Nicméně je nutné podotknout, že technologie implementačně závislé a nezávislé se velice často překrývají. V současné době je například vidět velice výrazně snahu, aby se technologie *Active Server Pages (ASP)* původně určená pouze pro *Internet Information Server (IIS)* firmy Microsoft stala nezávislou na platformě, podobně jako jiné technologie určené nejprve například pro server *Apache*.

Z toho důvodu jsem nakonec na rozdíl od let předešlých opustil standardní dělení serverových technologií na implementačně závislé a nezávislé a zvolil dělení podle platform různých úrovní. Musím však opět upozornit na skutečnost, že se rozhodně nejedná o úplný přehled možných technologií, ale pouze o přehled některých mnou vybraných technologií.

2.1 Projekt Apache

Původně se projekt *Apache* týkal pouze HTTP serveru vyvinutého pro platformu UNIXu, zejména Linuxu. V současné době se projekt Apache ambiciózně rozrůstá do mnoha směrů.

2.1.1 Projekt Apache Server

Je původním projektem projektu Apache. Poslední verze serveru Apache je verze 1.3, která obsahuje některé významné změny oproti starším verzím. Jedná se o podporu dynamických knihoven (*Dynamic Shared Object – DSO*), podporu pro *Windows NT/95*, experimentální podporu *NetWare 5.x* a mnoho dalších více či méně významných vylepšení. Mezi novinky teprve připravované verze 2.0 pak patří využití POSIX kompatibilních threadů na platformě UNIX, vylepšená podpora neUNIXových platforem jako například BeOS, OS/2 a Windows. Součástí projektu Apache server je také několik zajímavých nekomerčních projektů:

- *ActiveScripting for Apache (OpenASP)*
Je projekt, který si klade za cíl umožnit využívání technologie Active Server Pages (ASP) pro Apache Web server na platformě Windows.
- *Apache::ASP*
Je projekt emulace ASP na platformě Apache Web serveru pomocí perlovského modulu běžícího za pomoci mod_perl.

2.1.2 Perl Apache projekty

Perl Apache projekt podobně jako Java Apache projekt umožňuje rozšiřování vlastností a možností Apache serveru pomocí jazyka Perl. Nejvýznamnější projekty z našeho hlediska jsou následující.

- *Mod_perl project*
Tento projekt umožňuje pro server Apache psát moduly v jazyce Perl a tak využívat velké množství pro tento programovací jazyk již vytvořených knihoven.
- *HTML::Embperl*
Jedná se o projekt technologie podobné technologiím ASP či SSJS, založený však výhradně na skriptování v jazyce Perl.
- *HTML::Mason*
Jedná se o podobnou technologii, jako je technologie předešlá.

2.1.3 Java Apache projekty

Jedná se o celou řadu projektů založených na technologiích souvisejících s využitím jazyka Java a s Javou souvisejících technologiích, z nichž nejvýznamnější jsou snad následující projekty.

- *Mod_Java Project*
Jedná se o projekt modulu pro server Apache, který umožňuje pro tento server psát moduly v jazyce Java a tak pochopitelně využívat výhody tohoto jazyka (bezpečnost, přenositelnost) a velké množství knihoven pro tento jazyk již vytvořených.
- *Apache Jserv Project*

Je implementace *Java Servlet API 2.0* specifikace, spolupracující s Apache serverem prostřednictvím *mod_jserv* modulu.

- *Apache JSSI Project*
Jde o servlet umožňující dynamický *include* výstupu servletů na straně serveru. Jedná se o jakousi obdobu *appletů* na straně klienta. Servlety jsou do HTML dokumentů včleněny prostřednictvím tagu `<SERVLET>` stejně, jako je tomu na JavaSoft Java Web Server.
- *PicoServer*
Tento projekt si klade za cíl vytvoření odlehčeného HTTP 1.0 kompatibilního serveru, realizovaného 100% v jazyce Java. Tento server je určen pro zabudování do nejrůznějších zařízení.

2.1.4 Projekt Jakarta

Projekt *Jakarta* je otevřený projekt komerční kvality pro serverová řešení založená na platformě jazyka Java. Přestože je součástí projektu Apache, není určen výhradně pro Apache server, ale naopak se jedná o multiplatformní projekt. Tento projekt je společně řešen týmem projektu Apache a týmy firem SUN a IBM. Skládá se ze tří základních projektů.

- *Ant*
Je na platformě Javy založený nástroj typu *make*. Je rozšiřitelný pomocí Javovských tříd a využívá konfigurační soubory ve formátu XML.
- *Tomcat*
Je základní část celého projektu. Jedná se o referenční implementaci *Java Servlet 2.2 API* a *JavaServer Pages 1.1* technologií. Implementační tým je složen ze členů *Apache Jserv* týmu a odborníků firem IBM a Sun.
- *Watchdog*
Je prostředek pro validaci uživatelských produktů na shodu s *Java Servlet 2.2 API* a *JavaServer Pages 1.1* specifikacemi.

2.1.5 XML Apache projekt

Podobně jako *Jakarta* projekt jedná se v případě XML Apache projektu o projekt nezávislý na platformě, čili není omezen na spolupráci pouze s Apache serverem. Tento projekt má následující dílčí projekty.

- *Xerces*
Jde o projekt XML parserů v jazycích Java, C++ a Perl. Parsery jsou kompatibilní s W3C XML, DOM (Level 1 a Level 2) a SAX (version 2) specifikacemi.
- *Xalan*
Jedná se o XSLT stylesheet procesor v jazycích Java a C++. Plně podporuje W3 XSLT doporučení.
- *Cocoon*
Jde o ambiciózní projekt na XML založeném webovém publikování implementovaný zcela v jazyce Java.
- *FOP*
Jde o projekt tiskového formátoru pro XSL formátované objekty. Vstupem formátoru je dokument ve formě XML/XSL dokumentu a výstupem je dokument ve formátu PDF.

2.2 PHP/ZEND projekt

PHP projekt je samostatný projekt, který vznikl za účelem vytvoření technologie podobné technologiím ASP či SSJS. Primárním webovým serverem, pro který byla technologie PHP určena, byl Apache server. PHP se dočkalo značného rozšíření a v současné době je ve verzi 3.0. Netrpělivě očekávaná verze 4.0, která je v současné době ve stádiu neveřejných beta testů, přináší celou řadu revolučních změn. Jednak bude technologie PHP integrována s technologií ZEND. Technologie ZEND byla původně vytvořena jako nezávislý produkt kompatibilní s PHP, ale na rozdíl od PHP využívající run-time kompilaci PHP skriptů a tím umožňující jejich vyšší výkon. Nejvýznamnější nové vlastnosti jsou:

- Integrace se *ZEND*.
- Nezávislost na platformě. Zatím jsou podporovány pouze platformy Apache a IIS.
- Integrovaná podpora *session*.
- Využití dynamických knihoven na platformě UNIX.
- Změna konfiguračních souborů.

Firma *Zend Technologies* mimo samotnou *Zend Engine* dodává další produkty rozšiřující možnosti technologie PHP. Jedná se o produkt *Zend Cache*, který umožňuje kompilované skripty uchovávat v paměti *cache* a tím opět zvyšovat výkon celé technologie. Produkt *Zend Compiler*, který umožňuje PHP stránky kompilovat a tím chránit jejich obsah před okopírováním, a produkt *Zend Optimizer*, který optimalizuje již přeložený kód a tím opět umožňuje zvýšit výkon celého systému.

2.3 Internet Information Server (IIS) 5.0

Společně s Windows 2000 firmy Microsoft se na veřejnost dostala i nová verze *Internet Information Server (IIS) 5.0*. Windows 2000 také obsahují nové verze jazyků VBScript a JScript a dále novou verzi databázové knihovny MDA 2.5, která obsahuje integrovanou podporu XML. Nová verze IIS dále obsahuje četná další vylepšení, včetně vylepšení samotných ASP. Nejvýznamnější nebo snad nejzajímavější z těchto vylepšení jsou následující.

- *Server.Transfer* a *Server.Execute* metody
Tyto metody umožňují přenést zpracování jedné ASP stránky na druhou (*redirection*). Na rozdíl od běžně používané metody *Response.Redirect* se tyto metody realizují na straně serveru se zachováním kontextu stránky, ze které se volají.
- *AspError Object*
Pro snadnější uživatelské obsluhu chybových hlášek serveru je k dispozici nový objekt *AspError*, který umožňuje soustředit ošetření chyb do jednoho místa a na základě informací v tomto objektu obsažených generovat stránku vrácenou prohlížeči uživatele.
- *Skriptovací jazyky jsou ve verzi 5.0*
Je tedy možné při psaní kódu ASP využívat všechna vylepšení této verze jazyků VBScript a JScript. Například můžeme využít příkazu *with* v jazyce VBScript nebo příkazu *try-catch* v jazyce JScript.

- *Využití MDAC 2.5 integrace s XML*
Pomocí nové knihovny MDAC 2.5 je nyní možné data uložená v relační databázi ukládat či zpracovávat ve formě XML.

Je také nutné upozornit na nové chování ASP aplikací s novou verzí IIS. Tak například *ASP buffering* je nyní předvolen za defaultní, a proto bude nutné staré aplikace poopravit tak, aby zůstala s novou verzí IIS zachována jejich funkcionální. Takových změn chování nového IIS je více, pochopitelně jen některé z nich se mohou v dané konkrétní aplikaci projevit jako rozhodující, čili vyžadující změnu aplikace či nastavení IIS. Nová verze IIS také umožňuje nastavit různé parametry IIS, které v předchozích verzích nebylo možné nastavit, a tak lépe přizpůsobit výkon a chování IIS skutečné zátěži a prostředí, v němž je IIS provozován.

Celkově lze novou verzi IIS hodnotit pozitivně. Je zřejmou odpovědí na oprávněné požadavky programátorů ASP a správců IIS, kterým tato funkcionální ve starších verzích IIS a ASP podstatně chyběla.

2.4 JavaServer Pages (JSP)

JavaServer Pages jsou webová skriptová technologie podobná technologii Netscape *Server-Side JavaScript (SSJS)* nebo Microsoft *Active Server Pages (ASP)*. Krátký úvod do této technologie jsem již na tomto semináři v minulých letech uskutečnil. Nicméně, protože se jedná o jednu ze zásadních technologií současné a zřejmě i budoucí doby, aspoň dle mého názoru, nebude na škodu k této technologii i v tomto ročníku opět obrátit pozornost.

Podobně jako jiné skriptovací technologie i technologie JSP vkládá do stránek HTML kód programovacího jazyka. Na rozdíl od jiných technologií, které do stránek HTML vkládají jazyk, který je interpretován (JavaScript, Basic, Perl či PHP), klade si technologie JSP za cíl vkládat do stránek HTML jazyk Java, který je kompilován. Pravda, výsledkem kompilace je přenositelný *bytecode*, který je interpretován *Java Virtual Machine (JVM)*, ale tato skutečnost není z našeho hlediska podstatná. Technologie JSP řeší problém kompilace, podobně jako technologie SSJS. Při prvním přístupu na stránku JSP se tato stránka zkompiluje do formy *servletu* a uloží do cache takto zkompilovaných stránek a teprve v dalším kroku se zkompilovaná stránka spustí a výstup činnosti takto zkompilované stránky je odeslán klientu jako odpověď na jeho dotaz. Jak je vidět, technologie JSP s výhodou využívá o trochu starší technologii *servletů*. Co je to *servlet*? Zjednodušeně řečeno, podobně jako *applet* je kousek Javovského kódu, který se spouští v kontextu prohlížeče, je *servlet* kousek Javovského kódu, který se spouští v kontextu webového serveru.

Technologie JSP považují za velice zásadní i z následujícího důvodu. Pokud totiž na serveru spustím kód v jazyce Java, mám okamžitě k dispozici i všechny další výhody jazyka Java. Zejména naprostou nezávislost na platformě, jak hardwarové tak softwarové a mohu tudíž energii vloženou do vývoje aplikace mnohonásobně zhodnotit. Dále mohu pro standardní přístup k databázovým systémům využít technologii *Java Database Connection (JDBC)* a technologii *Enterprise JavaBeans (EJB)* pro transakční zpracování v oblasti obchodní logiky. Pochopitelně obě tyto technologie jsou též založeny na jazyce Java a jsou tudíž na platformě nezávislé.

Marně si kladu otázku: proč se tyto technologie více nevyužívají? A napadá mne jediná odpověď. Musí se totiž nainstalovat. Zatímco řešení založené na IIS a ASP se snadno nainstaluje spolu s instalací operačního systému Windows NT.

Příklad č.1

```
<HTML>
<HEAD><TITLE>JSP Page - 1</TITLE></HEAD>
<BODY>
<jsp:useBean      ID="ssb"
                  SCOPE="session"
                  CLASS="javax.spl es. SuperSimpleBean"
/>
<jsp:setProperty NAME="ssb"
                  PROPERTY="technologie"
                  VALUE="JSP"
/>
<h2>Tato stránka používá technologii :
    <jsp:getProperty NAME="ssb" PROPERTY="technologie" />
</h2>
</BODY>
</HTML>
```

Technologie JSP dovoluje pracovat buď přímo s EJB pomocí speciálních značek (*tag*), nebo vnořovat Javovský kód přímo do HTML stránky. Pokusím se váženému čtenáři vysvětlit obě tyto možnosti pomocí velice jednoduchých příkladů. V prvním příkladu ukáži práci s velice jednoduchým EJB pomocí značek. Tento EJB je opravdu velice jednoduchý, obsahuje totiž pouze jedinou *property*, kterou budeme inicializovat určitou hodnotou a její hodnotu poté zobrazíme.

První značka *jsp:useBean* je v našem příkladu určena pro deklaraci ve stránce používaného EJB, kterému je přidělen identifikátor *ssb*, pomocí kterého se na něj dále budeme v našem příkladu odkazovat. Dále je mu přidělen rozsah platnosti *session*, který znamená, že vytvořený EJB bude existovat po celou dobu tzv. *session* (obdobu *session* v ASP). Dále je zde informace, o jaký EJB se vlastně jedná ve formě syntaxe obvyklé ve světě Javy.

Značka *jsp:setProperty* uloží do *property* se jménem *technologie* hodnotu *JSP*. Identifikace správného EJB se určí parametrem *NAME*, který má hodnotu *ssb*. Značka *jsp:getProperty* je určena pro vyzvednutí hodnoty z EJB. Současně se vyzvednutá hodnota vloží do stránky HTML namísto značky *jsp:GetProperty*.

Druhý příklad bude opět velice jednoduchý. Pomocí JSP vypíšeme na stránku aktuální čas a tuto stránku vrátíme klientu. Nejprve uložíme aktuální čas do proměnné *dt*, která je typu *Date*. Všimněme si, že současně proměnnou vytvoříme pomocí konstruktoru *new*. Tato část programového kódu je uzavřena mezi značky *<%* a *%>*. Hodnotu proměnné *dt* po transformaci na hodiny, minuty a sekundy vypíšeme pomocí značek *<%=* a *%>*. Je vidět, že obě dvojice značek mají stejný význam, jako u technologie ASP.

Příklad č.2

```
<HTML>
<HEAD><TITLE>JSP Page - 2<TITLE></HEAD>
<BODY>
<% java.util.Date dt =
new java.util.Date(System.currentTimeMillis());
%>
<H1>Time: </H1>
<%=dt.getHours()%>:
<%=dt.getMinutes()%>:
<%=dt.getSeconds()%>
</BODY>
</HTML>
```

Při srovnání technologie JSP a ASP si jistě každý případný uživatel JSP položí otázku, jakou mají skutečnou podporu. Za technologií ASP stojí firma Microsoft se svojí váhou největší počítačové firmy na světě, ale jak je to s technologií JSP? Odpověď je jednoduchá. Za technologií JSP stojí firmy SUN, IBM, team Apache a mnohé další firmy, například i firma Netscape. Další otázkou může být například otázka, co je všechno třeba k tomu, abych technologii JSP mohl vůbec provozovat? Odpověď je velice jednoduchá. Jsou třeba následující komponenty.

- *Java Development Kit (JDK)*
Hlavní součástí JDK je Java Virtual Machine (JVM), která umožňuje běh programů napsaných v jazyce Java. V současné době existují následující nejznámější implementace.
 - SUN JDK – oficiální implementace od tvůrců jazyka Java pro nejznámější hardwarové platformy (Intel, Unix, atp.)
 - IBM JDK – implementace od firmy IBM, mimo jiné pro AIX, OS/2, OS/390, OS/400, VM/ESA
 - Blackdown Linux – implementace pro i386 Linux
 - Japhar – jednoduchá implementace
 - Kaffe OpenVM – nezávislá implementace, nejen pro Linux
 - SGI IRIX Java – implementace firmy SGI pro jejich počítače
 - TowerJ – moderní implementace (třetí generace)
 - Apple MRJ – implementace pro počítače Apple
 - MS SDK for Java – implementace od firmy Microsoft
- *Java Servlet Development Kit (JSDK)*
Programové prostředí implementující rozhraní Java Servlet API, nutné pro chod JSP a samotné JSP. Existují opět různé implementace.
 - SUN JSDK – implementace od firmy SUN
 - Projekt Jakarta (TomCat) – referenční implementace od teamu Apache a firmy SUN
 - Mac JSDK – neoficiální verze pro Mac
 - Jserv – implementace pro server Apache
 - LiteWebServer – od Gefion software
 - Java Web Server – od firmy SUN
 - Jigsaw – od Web Consortia

- WebSphere – od IBM
- Jrun – od Allaire
- celá řada dalších, především k volnému použití

Ve skutečnosti není výše uvedený seznam zcela přesný. Některé produkty jsou opravdu ekvivalenty JSDK, jiné představují webové servery se zabudovaným JSDK. Další jsou zase doplňky k normálním serverům (*servlet runner*) a umožňují na nich provoz servletů a JSP.

Přestože ve své podstatě JDK a JSDK postačují pro provoz JSP, budete ve většině případů však často požadovat i dva následující produkty.

- *Java DataBase Connection (JDBC)*
Je rozhraní, které umožňuje z prostředí jazyka Java přistupovat k databázím. Jedná se ve své podstatě o speciální drivery, které realizují toto rozhraní pro jednotlivé databázové systémy. Součástí JDK je však univerzální *JDBC-ODBC* driver, který umožňuje pro přístup k databázím používat rozhraní ODBC, které je součástí Windows firmy Microsoft a pomocí něhož je opět možné přistupovat k různým databázovým systémům nejrozličnějších výrobců.
- *Enterprise JavaBeans (EJB)*
Je komponentní, transakční, distribuované aplikační architektura. Analogií k EJB je například *CORBA ORB* nebo *COM+* firmy Microsoft. Jedná se o architekturu umožňující vytvářet tzv. *aplikační servery*, které realizují tzv. *obchodní logiku* celé aplikace v tzv. *třívrstvé architektuře*.

Závěr

Přestože se mi v tomto příspěvku nepodařilo podchytit celý bohatý vývoj intranetové problematiky v období od minulé konference, snad bude i takový příspěvek pro čtenáře vítaným informačním přínosem. V celém vývoji intranetových technologií vidím dvě výrazné tendence.

- Standardizace, nezávislost na platformě, XML, Java, servlety, JSP, JavaScript
- Firemní řešení, Microsoft, XML, IIS a ASP, COM+, VBScript, JScript

Je na samotném uživateli, pro které řešení se rozhodne a za jakou cenu. Přestože by se mohlo zdát, že intranetová řešení jsou velice rozvinutá a jejich technologie je pečlivě propracovaná, není tomu tak. Bohužel velická slabina spočívá stále v oblasti vývojových nástrojů, které by umožnily kvalitní vizuální návrh stránek na jedné straně a programování těchto samých stránek na straně druhé. Nejenže je takových nástrojů velice málo, ale ve své podstatě každý z těchto nástrojů představuje své vlastní jedinečné řešení, protože v této oblasti, snad mimo *JavaScript Beans* firmy Netscape, neexistují žádné obecně přijaté standardy.

Závěrem svého článku se chci omluvit čtenáři za části příspěvku, které navazují na mé předešlé příspěvky či předpokládají čtenářovi znalosti určité problematiky, a proto mohou být poněkud méně srozumitelné.

Odkazy

MSIE 5.5	http://www.infinisource.com/features/internet-explorer55.html
Gecko	http://developer.netscape.com/software/communicator/ngl/index.html
Mozilla	http://www.mozilla.com
XHTML	http://www.w3.org/TR/xhtml1/
HTML Tidy.....	http://www.w3.org/People/Raggett/tidy/
Apache	http://www.apache.org/
Apache Server.....	http://www.apache.org/httpd.html
Perl Apache projekt.....	http://perl.apache.org/
Java Apache projekt.....	http://java.apache.org/
Jakarta projekt.....	http://jakarta.apache.org/
XML Apache projekt	http://xml.apache.org/
PHP projekt.....	http://www.php.net/
Zend projekt.....	http://www.zend.com/
IIS 5.0.....	http://www.microsoft.com/mind/0499/iis5/iis5top.htm
IIS 5.0.....	http://msdn.microsoft.com/workshop/server/asp/server02282000.asp
JSP.....	http://java.sun.com/products/jsp/
JavaBeans.....	http://java.sun.com/beans/
Servlety	http://java.sun.com/products/servlet/
JDBC.....	http://java.sun.com/products/jdbc/
EJB.....	http://java.sun.com/products/ejb/
Servlets Taverne.....	http://www.interpasnet.com/JSS/
JSP v cesku	http://jsp.ami.cz/
Molhanec.....	http://martin.feld.cvut.cz/~mmm

Ing. Martin Molhanec, CSc

ČVUT-FEL, K-313
Technická 2
166 27 PRAHA 6, Dejvice
tel.: ++420 (2) 2435 2118
fax: ++420 (2) 2435 3949
email: molhanec@fel.cvut.cz

Molhasoft
Křivenická 406
181 00 PRAHA 8, Čimice
tel.: ++420 (2) 855 05 21
fidonet: 2:420/12.102
email: molhanec@technologist.com