

Technologie tvorby intranetových aplikací.

MARTIN MOLHANEC

Katedra elektrotechnologie
Elektrotechnická fakulta, České vysoké učení technické
Praha, Česká republika

Abstrakt

Tento příspěvek se zabývá problematikou tvorby intranetových aplikací. V příspěvku jsou stručně popsány hlavní technologické možnosti vytváření dynamických HTML stránek, jak na straně klienta, tak na straně serveru. Dále jsou popsána rozhraní CGI, FastCGI, ISAPI a některá další. Je také popsána problematika vývojových prostředků IDC/HTX a ASP. Cílem článku je vytvoření přehledu všech současných možností tvorby intranetových aplikací.

1. Úvod

Podobně jako před několika lety, byla odborná infromatická veřejnost vzrušena nástroji CASE nebo objektově orientovaným paradigma, je dnes jedním ze žhavých témat využití internetových technologií v prostředí informačních systémů (dále jen IS), pro které používáme název intranet. Pokusím se ve svém příspěvku přinést do dané problematiky trochu světla. Nepředpokládám, že bude daná problematika vysvětlena do nejmenších detailů. Není to možné, jak s ohledem na rozsah tohoto příspěvku, tak na velký a neustále narůstající obsah dané problematiky.

Především je nutné si uvědomit, že přes svoji novost, vyrůstá daná problematika z předešlých vývojových kroků učiněných v informatice a do určité míry se dá někdy říci, že jde o „staré“ v „novém“ balení. Když se podíváme do historie rozsáhlých informačních systémů, byly před příchodem osobních počítačů používány především terminálové systémy spojené s mainframe počítačem. Tyto systémy, však postupně byly nahrazovány systémy založenými na bázi osobních počítačů, zejména typu IBM PC, sítě LAN a technologií klient/server. Tyto nové systémy byly pro laické uživatele atraktivní, velice často umožňovaly lokální činnost bez sítě LAN, přitom malé informační systémy založené na programech typu dBase nebo Paradox měli daleko snadnější ovládání než

těžkopádné klávesy terminálu nemluvě o češtině. Pozdější využití osobních počítačů se opět vrací k centralizaci s výkonným souborovým serverem nebo databázovým serverem. Navíc uživatel počal využívat velice atraktivní prostředí Windows. Bohužel výše zmíněnou centralizací se opět ke slovu dostává správce výpočetních prostředků, zejména správce sítě LAN. Končí doba, kdy si uživatel vystačil a své PCčko zvládl bez cizí pomoci. Stačí ovšem jedna věta: „síťová instalace Windows a jejich aplikací“. Ten kdo rozumí, tomu nemusím napovídat, kdo nerozumí, doporučuji vyzkoušet a v čim více heterogenním prostředí, tím lépe, aspoň pochopí.

Technologie intranetu, přináší řešení. Na stanici uživatele stačí nainstalovat Windows (bohužel) s jedním internet prohlížečem (Netscape Navigator nebo Microsoft Explorer) a je vystaráno! Všechny informační aplikace intranetového charakteru se instalují na síťový server a jsou okamžitě přístupné všem uživatelům. Svým způsobem je možné i říci, že toto řešení je podobné starým řešením se sítí terminálů a mainframe počítačům. Starý textový terminál byl nahrazen standardním (na bázi HTML jazyka) kvalitním multimediálním grafickým terminálem a mainframe počítač je nahrazen výkonným aplikačním serverem. Správce výpočetní techniky se opět stará o aplikace instalované na aplikační server a nemusí zápasit s problémy na desítkách uživatelských počítačích. Není se tedy třeba intranetové technologie nikterak obávat, jedná se sice o novou kvalitu, ale její filozofie se dá uchopit a využívat. Podle mého přináší nové výhody, ale pochopitelně jako každá nová skutečnost, která se ještě vyvíjí, má doposud své problémy a nedostatky.

Technologie intranetu vychází s technologie internetu. Technologie internetu je postavena na standardním protokolu TCP/IP, určeném zejména pro síť WAN, a využívání aplikačních protokolů postavených nad touto síťovou vrstvou. Těchto protokolů je celá řada, jmenujme si například protokoly telnet, ftp, gopher, nfs a mnohé další. Ale teprve použití protokolu http přenášejícího stránky napsané v jazyce html, koncepce jejich prohlížeče (Internet Browser) a WWW serveru (httpd server), dohromady tvořící to čemu dnes jednoduše říkáme Web, vytvořilo rychle šířící se technologie, která aplikovaná na průmyslové systémy vytvořila pojem *INTRANET*.

Jaké jsou hlavní aspekty technologie intranetu z hlediska jeho tvůrce? Jsou pochopitelně stejné jako u všech informačních technologií. Jedná se o následující aspekty:

- Komunikace s uživatelem (formuláře, meny, volby, výběry, grafy, atp.)
- Logika aplikace (programovací jazyk a podobné prostředky)
- Uložení dat (databáze)

Zkusme se nyní na technologii intranetu podívat z hlediska výše uvedených aspektů.

- *Komunikace s uživatelem* – je definována pomocí HTML stránek, které jsou interpretovány na počítači klienta internet prohlížečem
- *Logika aplikace* – je určena vytvářením dynamických stránek na WWW serveru, na základě informací získaných klienta

- *Uložení dat* – je stejné jako u ostatních informačních systémů, s databází komunikuje program spouštěný WWW serverem při vytváření dynamických stránek

Je zřejmé, že objektem našeho zájmu budou *prostředky na straně klienta a na straně serveru*, protože aspekt ukládání dat zůstává stejný jako jiných informačních systémů, které nejsou založeny na intranetové technologii.

2. Strana klienta

Tvůrce intranetových aplikací má na straně klienta celou řadu prostředků, které může použít při vývoji své aplikace. Původní HTML stránky byly převážně statické. Postupem doby však získal celou řadu možností, jak přiblížit klasickou stránku HTML formuláři IS aplikace. Toto obohacení je založeno jednak na rozšíření jazyka HTML o nové vhodné prvky, dále o možnost využití tzv. objektů vložených do HTML stránek a v poslední řadě o tzv. dynamické HTML využívající skriptovacích jazyků na straně klienta.

2.1. HTML (Hyper-Text Markup Language)

Vlastní jazyk HTML je založen na obecněji založeném jazyku SGML (Standard Generalized Markup Language). Účelem tohoto jazyka je definovat vzhled dokumentu nezávisle na zobrazovacím zařízení. Formátování dokumentu se provádí na základě tzv. TAGů, jedná se o domluvené značky, které určují jaké formátování bude mít určitá část dokumentu. Například, následující řádka

Jmenuji se *Martin Molhanec* a pracuji na **ČVUT**.

se jazykem HTML napíše následujícím způsobem:

```
<CENTER>Jmenuji se <I>Martin Molhanec</I> a pracuji na  
<B>ČVUT</B></CENTER>
```

Nicméně není účelem tohoto příspěvku popisovat jazyk HTML, ale upozornit na jeho aspekty, které jsou vhodné pro tvorbu IS aplikací. Nebudu tedy v tomto příspěvku hovořit ani o tak důležitých rysech jazyka HTML jako jsou například *rámy*, *kaskádní styly* nebo *klikací mapy*. Vývojář IS aplikace se bez nich může obejít, zmíním se ale stručně o následujících dvou rysech, bez kterých se žádný vývojář IS aplikace obejít nemůže.

2.1.1. Tabulky

Tabulky umožňují vývojáři IS zobrazovat data v tabelární formě. Jedná se zejména o výstupy z databází. Bez tohoto prvku, by byly stránky HTML pro tvůrce IS aplikací prakticky nepoužitelné. Do jednotlivých políček tabulky je možné umísťovat nejen texty, ale i grafiku. Ukázka takové tabulky, získaná jako výstup z databáze pomocí technologie *Active Server Pages*, včetně vložených obrázků je na obr. 1.

2.1.2. Formuláře

IS aplikace bez vstupní dat by snad ani nebyla žádná aplikace. Jazyk HTML nabízí pro vstup dat možnost použití tzv. formulářů. Formuláře jsou omezeny značkami <FORM> a </FORM>. Mezi tyto značky se vkládají značky vstupních prvků, které pokrývají všechny základní vstupní elementy použitelné v IS aplikacích. Jedná se o následující prvky:

- vstupní pole <INPUT>
- textové pole <TEXTAREA>
- výběrové pole (rozbalovací nabídka) <SELECT>
- tlačítko <BUTTON>

Všechny výše uvedené značky mají možnost doplnění o různé parametry (*options*), které umožňují jejich rozmanitost. Příklad formuláře s různými typy vstupních elementů je na obr. 2.

2. Vložené objekty

Protože jazyk HTML přes svoji veškerou rozmanitost nedokáže vytvořit všechny elementy, které by snad vývojář mohl ve své aplikaci požadovat. Byl jazyk HTML rozšířen o možnost vkládat do stránek prvky vytvořené nikoliv prostředky jazyka HTML, ale mimo něj. Důležitá je ovšem skutečnost, že tyto elementy se zobrazují jako část celé zobrazované stránky a jsou její integrální součástí. To znamená, že příkazy jazyka HTML pro zarovnání a velikost elementů HTML jazyka platí stejně pro vložené objekty.

2.2.1. Java applety

Java applety jsou objekty vytvořené v jazyce Java (přesněji řečeno napsané v jazyce Java a přeložené do binární formy) a vložené do stránek HTML. Jazyk Java byl vyvinut firmou Sun, jako multiplatformní jazyk podobný jazyku C++, s vyšší bezpečností a jednoduchostí. Jazyk je multiplatformní, to znamená, že zdrojový kód Javy není přeložen do strojového kódu procesoru, kde bude aplikace běžet, ale je přeložen do speciálního tzv. *byte code*, který je na konkrétním procesoru nezávislý. Na systému na kterém potom bude aplikace napsaná v jazyce Java běžet musí být potom k dispozici tzv. *Java Virtual Machine* (dále jen JVM), která bude přeložený program v jazyce Java vykonávat (interpretovat). Programy v jazyce Java po svém přeložení mohou běžet na libovolném procesoru a operačním systému, musí být k dispozici pouze JVM pro danou platformu. Toto je z hlediska tvůrce IS aplikací nesmírná výhoda. Na straně druhé je ovšem výsledná aplikace v jazyce Java pomalejší nežli podobná aplikace napsaná například v jazyce C a přeložená přímo do strojového kódu. Nicméně, pokud se podíváme na typické IS aplikace, vidíme, že většina z nich se skládá hlavně ze vstupu dat a databázových dotazů. Vzhledem k tomu, že Java applety budou zřejmě nejvíce využívány na speciální elementy vstupu nebo výstupu dat, lze předpokládat, že menší výkon jazyka Java nebude zřejmě na závadu.

Java applet je přeložený program v jazyce Java, který je uložen na WWW serveru, společně s HTML stránkami. Pokud HTML stránka obsahuje odkaz na Java applet, bude tento applet podobně jako HTML stránka stažen do počítače klienta a jeho kód bude vykonán JVM na počítači klienta v rámci prostředí HTML prohlížeče, respektive v rámci stažené HTML stránky. Jedna stránka HTML může obsahovat více Java appletů, které jsou zobrazovány v rámci stránky a jsou její integrální součástí.

Ukázka Java appletu, který vytváří *stromový navigátor* určený pro snadnější procházení po WWW aplikaci je na obr. 3. Zdrojový kód HTML stránky do které je tento applet vložen je na příl. 1. Další možnosti použití Java appletů jsou například:

- Zobrazování grafů
- Zabudovaná kalkulačka
- Komplikovaný formulář, pokud nestačí standardní možnosti HTML
- Animace

Výhodou Java appletu je, že se jeho kód vykonává na straně klienta, nezatěžuje tedy síťové spojení a také odezva je téměř okamžitá. Nevýhodou je ovšem skutečnost, že se celý kód appletu musí na klienta přenášet z WWW serveru. Pochopitelně, vzhledem k tomu, že klient přenesené stránky, včetně appletů kešuje na straně klienta, znamená to, že se kód appletu přenáší ke klientovi pouze jednou.

Další velká výhoda jazyka Java je skutečnost, že jeho syntaxe je založena na ověřeném objektově orientovaném programovacím jazyce C++. Na rozdíl od jazyka C++ je však daleko bezpečnější, protože neobsahuje ukazatele (způsobující snad nejvíce chyb, které programátoři v jazyce C a C++ udělají). Obsahuje naopak výkonnou správu paměti, která se vykonává automaticky při vzniku a zániku objektů a vyspělou obsluhu chybových událostí.

Dalším prvkem bezpečnosti je vlastnost, že Java applety nemají žádnou možnost přistupovat k souborovému systému klientského počítače. Není tedy možné vytvářet *viry* na základě Java appletů.

Java applety dále umožňují po síti komunikovat prostřednictvím protokolu TCP/IP. Tato vlastnost umožňuje například, aby Java applet z klientského počítače komunikoval přímo s WWW serverem, například s databázovým serverem, který je na též serveru v provozu. Bezpečnost je v tomto případě založena na skutečnosti, že applet může navázat spojení pouze ze serverem, ze kterého je stažen do klientského počítače.

2.2.2. Active-X

Objekty Active-X byly vyvinuty firmou Microsoft, jako reakce na vznik Java appletů. Jejich použití je podobné. Na rozdíl od Java appletů se jedná defakto o tzv. OLE objekty, které firma Microsoft vyvinula v rámci svého operačního

systemu Windows. OLE objekty, potažmo tedy Active-X objekty mohou být napsány v libovolném programovacím jazyku, který jejich vytváření podporuje, například C++, Visual Basic, Delphi, Visual Objects, a další.

Využití Active-X objektů je v zásadě stejné jako u Java appletů. Jako příklad si uvedeme *stromový navigátor*, tentokrát jako objekt Active-X. Jeho vzhled je na obr. 4 a příslušný zdrojový soubor HTML stránky ve které je uložen je na příl. 2

Podstatnou nevýhodou Active-X objektů je skutečnost, že nejsou multiplatformní! Po přeložení, mohou být vykonávány pouze na procesoru pro který byly přeloženy. Navíc musí klientské prostředí obsahovat technologii OLE objektů, což je zatím možné pouze v rámci operačních systémů firmy Microsoft. Firma Microsoft však v současné době plánuje zavést svoji OLE technologie i na systémy typu UNIX, nicméně se dá předpokládat, že to bude spojeno s mnoha těžkostmi, nehledě na určitý odpor ze strany uživatelů systémů UNIX.

Používání Active-X objektů znamená tedy pro uživatele určitá omezení:

- Objekt Active-X je použitelný pouze pro klienty s podporou firmy Microsoft.
- Objekt Active-X se musí pro každou platformou (procesor) znovu přeložit.

Objekty Active-X jsou navíc od Java appletů jsou daleko méně bezpečné. Spuštěná Active-X aplikace je omezená pouze možnostmi operačního systému. Je tedy možné psát a taky byly napsány Active-X viry. Navíc existuje nebezpečí, že špatně napsaný, například v C++, Active-X prvek zhroutí celý systém klienta.

2.3. *Dynamické HTML*

Podstata dynamického HTML na straně klienta spočívá ve využití tzv. *skriptových jazyků*, jejichž zdrojový kód je součástí HTML stránky a je vykonáván na straně klienta. Jednoduše řečeno, jedná se programový kód ve zdrojovém tvaru, který je na straně klienta prohlížečem HTML stránek interpretován. Hlavním účelem dynamického HTML na straně klienta, je koordinovat spolupráci mezi vloženými objekty (Java applety, Active-X objekty) a dalšími elementy HTML dokumentu, obsluhovat různé události (například stisknutí tlačítka, natažení nové HTML stránky) a *on line* generovat HTML dokument na straně klienta při jeho stažení.

Kód skriptových jazyků je v rámci HTML stránky omezen značkami <SCRIPT> a <SCRIPT/>. Samotný kód je pak ještě umístěn mezi značky HTML komentáře. Prohlížeče, které skriptový jazyk nepodporují, omezující značky podle pravidel HTML ignorují a samotný program skriptu je pro ně komentářem.

Skriptové jazyky mohou přistupovat pouze k objektům, které jsou součástí prohlížeče, včetně vložených objektů. Například mohou měnit jejich barvu, atp.

Mohou i nové objekty vytvářet, například nové okno prohlížeče, které natáhne novou stránku.

Z hlediska tvůrců aplikací IS je skriptový jazyk klienta zejména vhodný pro kontrolu vstupních dat ve formuláři. Pokud by nebylo možné skriptového jazyka využít, musely by se data ve formuláři přenést na WWW server a tam kontrolovat prostředky serveru, například programem v CGI nebo jiným způsobem. V případě chyby by musel server WWW vygenerovat novou stránku, kterou by vrátil klientovi. Tato činnost by zatěžovala, jak síť, tak server a navíc by odezva byla v mnoha případech poměrně dlouhá.

2.3.1. Javascript

Javascript byl vyvinut firmou Netscape, zejména pro ovládní a koordinaci Java appletů. Ostatně syntaxe tohoto jazyka je silně jazykem Java ovlivněna. Z hlediska uživatele je Javascript bezpečným jazykem, protože může přistupovat pouze k objektům prohlížeče. Nelze ho tedy zneužít pro vytváření virů. Jednoduchá ukázka programu v Javascriptu, který při stisknutí tlačítka vypíše oznámení o jeho stisknutí je na příl. 3. Obrázek tlačítka, které tuto událost aktivuje, jak je zobrazeno uživateli je na obr. 5.

2.3.2. JScript

JScript je varianta jazyka Javascript, který byl vyvinut firmou Microsoft. Důvod pro jiný název je ten, že firma Microsoft od firmy Netscape nezakoupila na jazyk Javascript licenci. Přestože firma Microsoft tvrdí, že její JScript je 100% kompatibilní s Javascriptem firmy Netscape, existují bohužel mezi oběma verzemi určité rozdíly a navíc existují rozdíly v implementacích pro různé verze prohlížečů. Tak například Jscript v MS Internet Exploreru verze 3.0 je kompatibilní pouze s Javascriptem v Netscape Navigator verze 2.0, a další rozdíly je možné nalézt i mezi dalšími verzemi obou prohlížečů.

2.3.3. VBscript

Skriptový jazyk VBscript je reakcí firmy Microsoft na jazyk Javascript firmy Netscape. Syntaxí jazyk VBscript vychází z populárního jazyka Visual Basic firmy Microsoft, je však velice silně omezen, což ovšem z hlediska jeho užití není na závadu. Na rozdíl od Javascriptu, který je firmou Microsoft podporován ve formě jazyka JScript, není naopak jazyk VBscript podporován firmou Netscape. Použití jazyka je stejné jako v případě Javascriptu. Ukázka příkladu v jazyce VBscriptu, který realizuje stejný příklad, jako již byl ukázán v části věnované Javascriptu je na příl. 3.

3. Strana serveru

Přes veškeré možnosti, které jsou realizovatelné na straně klienta nelze stále (pokud budeme předpokládat, že WWW server pouze dodává klientovi statické stránky) realizovat takovou běžnou záležitost, jako získat potřebné údaje z databáze. Pro tuto činnost je nezbytné, aby server spustil program, který bude

komunikovat s databází a na základě výsledků této činnosti sestrojí stránku, kterou server klientu vrátí. Je mnoho způsobů, jak tuto činnost realizovat.

3.1. Obecná rozhraní

Obecná rozhraní definují rozhraní mezi serverem a aplikací, která komunikuje s WWW serverem nezávisle na typu serveru, operačním systému a procesoru. Jejich výhodou je většinou snadná implementace a značné rozšíření. Nevýhody spočívají většinou v menším výkonu rozhraní a nemožnosti využití specifických vlastností toho či onoho WWW serveru.

3.1.1. CGI (*Common Gateway Interface*)

CGI bylo prvním rozhraním, které vzniklo za účelem rozšíření WWW serveru o možnost generovat dynamické stránky. Jeho princip je velice jednoduchý (obr. 6) a dá se vyjádřit následujícím algoritmem.

- Klient zašle serveru dotaz, jehož URL ukazuje na CGI skript. Ve většině případů URL obsahuje i parametry, které klient předává serveru.
- Server zjistí, že požadované URL ukazuje na CGI skript. Spustí tedy tento skript a předá mu parametry, potřebné k jeho činnosti.
- CGI skript vykoná svoji činnost (například dotaz do databáze), jejím výsledkem je vytvoření stránky HTML. Tuto stránku aplikace zašle na svůj standardní výstup.
- Server přijme stránku od CGI skriptu a pošle její obsah klientovi.

Výhody CGI rozhraní jsou zejména následující:

- Je snadno implementovatelné, protože se CGI skript spouští jako proces shellu
- Je implementováno defakto pro všechny WWW servery
- Parametry jsou předávány jako proměnné prostředí nebo na standardní vstup aplikace
- CGI skript může spustit program napsaný v libovolné jazyce
- Generovaná stránka se vrací serveru prostřednictvím standardního výstupu aplikace
- Vzhledem k tomu, že CGI skript neběží v kontextu serveru, nemůže jeho činnost narušit

Nevýhody CGI rozhraní jsou:

- Při každém vyvolání CGI skriptu se spouští shell, což je činnost náročná na čas i paměť
- Pokud CGI skript komunikuje s databází, musí do při každém spuštění opětovně přihlásit
- Poněkud obtížně se udržují informace, se kterými pracuje několik spuštění CGI skriptu po sobě

- Při mnoha současných požadavcích, mnoho spuštěných CGI skriptů značně zatěžuje systém
- CGI skripty nejsou zcela bezpečné, špatně napsaný skript může umožnit průnik do počítače

Přes veškeré výhody CGI rozhraní, mezi které lze počítač i skutečnost, že pokud je CGI skript napsán v některém interpretovaném jazyce, například Perl, je defakto přenositelný mezi všemi platformami (Unix, Windows, Netware), je jeho podstatnou nevýhodou to, že u velkých WWW systémů, které jsou zatíženy tisíci požadavky denně, nesmírně zatěžuje systém, výsledkem je pak podstatné zvýšení odezvy, čili času, který uplyne mezi požadavkem a jeho uspokojením. Další nevýhodou je poměrně složité zajištění existence informací, které by byly předávány mezi jednotlivými požadavky jednoho uživatele a sdíleny tedy víceným spuštěním CGI skriptu. Nicméně existuje obrovské množství CGI skriptů, které jsou prostřednictvím internetu volně dostupné a tím umožňují vývoji WWW aplikace je přímo využít nebo se jejich zdrojovým kódem inspirovat. Ukázka jednoduchého CGI skriptu, který vypíše uvítání a identifikace počítače na kterém běží je na příl. 4.

3.1.2. FastCGI

Rozhraní FastCGI vzniklo jako reakce na podstatné nevýhody CGI skriptů. O vývoj rozhraní FastCGI se stará firma Open Market. Princip FastCGI rozhraní spočívá ve skutečnosti, že skript je umístěn trvale v paměti a očekává požadavky od serveru, které okamžitě vykonává. Výsledkem je podstatné snížení zatížení serveru a zrychlení odezvy. Rozhraní FastCGI má další přednosti před tradičním rozhraním CGI.

- Je defakto kompatibilní s rozhraním CGI a má všechny jeho výhody
- Díky trvalému umístění skriptů v paměti méně zatěžuje server
- Snadnější sdílení informací mezi jednotlivými požadavky WWW serveru
- Umožňuje vykonávat CGI skripty na jiném počítači nežli běží WWW server
- Možnost tvorby filtru HTML stránky či ověření uživatele

Vysvětleme si nyní podrobněji výše uvedené výhody. První tři vyplývají přímo ze záměru tvůrců rozhraní FastCGI a jeho vlastností. Další si vyžadují vysvětlení. Pro provoz FastCGI rozhraní jsou potřeba dvě jeho základní části. Část, která je součástí WWW serveru a část, která je součástí FastCGI skriptu. Obě dvě části mezi sebou komunikují pomocí jednoduchého protokolu. Část, která je součástí serveru je většinou implementovaná na jeho interní rozhraní. V současné době existující servery, které podporují rozhraní FastCGI jsou následující:

- Netscape Commerce and Enterprise Servers
- Server NCSA
- Apache Server

- Stronghold Secure Webserver
- Open Market WebServer

Díky již zmíněné koncepci, komunikace dvou částí FastCGI rozhraní pomocí síťového protokolu, je možné, aby obě dvě části byly umístěny na rozdílných počítačích. Část, která je součástí klienta je k dispozici ve formě knihoven pro většinu běžných jazyků používaných pro psaní CGI skriptů. Jedná se například o následující jazyky: C, Perl, Java, Tk/Tcl. Jednou ze základních činností této knihovny je přeměrování standardního výstupu CGI skriptu do FastCGI protokolu. Ukázka jednoduchého CGI skriptu je na příl. 4. Na příl. 5 je potom jeho verze pro FastCGI. Na rozdíl od verze pro CGI, tento script navíc počítá počet přístupů. Je patrné, že přechod na fastCGI může být téměř rutinní záležitostí.

Nyní zbývá ještě vysvětlit poslední výhodu. Vzhledem k tomu, že je FastCGI rozhraní na serveru připojeno přímo na jeho interní rozhraní, je možné FastCGI skript volat i při dalších činnostech WWW serveru nežli při pouhém požadavku na spuštění CGI skriptu. Jedná se jednak o možnost tzv. filtru. Při této činnosti, uživatel zadá URL se speciálním jménem souboru, server potom spustí FastCGI skript a na standardní vstup (ve skutečnosti do protokolu FastCGI) mu tento soubor předá. FastCGI skript pak může obsah tohoto souboru modifikovat, například provést překódování češtiny nebo stránka může obsahovat vzor stránky (šablonu) a skript na základě tohoto vzoru a dotazu do databáze vygeneruje definitivní podobu stránky, kterou vrátí na standardní výstup (ve skutečnosti do protokolu FastCGI).

Další možností je ověření uživatele. V tomto případě FastCGI skript zjistí zdali daný uživatel může danou HTML stránku požadovat nebo ne. Mimo ověření uživatele a možnosti vytváření filtrů se uvažuje o dalších možnostech, které v budoucnu může protokol FastCGI podporovat.

Ještě na jednu podstatnou výhodu FastCGI je nutné upozornit. Protože je skript FastCGI neustále v paměti, může v případě komunikace s databází udržovat toto spojení stále otevřené a tím velice podstatně snížit odezvu FastCGI skriptu při komunikaci s ní. Další výhodou může být pak snadná existence proměnných, které existují během trvání FastCGI aplikace.

3.1.3. Java Servlets

Toto rozhraní bylo vyvinuto firmou Javasoft (součást firmy Sun, tvůrce jazyka Java) původně pro server Java Web Server, který se vyznačoval vlastností, že byl celý implementován v jazyce Java. Nicméně již od počátku je toto rozhraní koncipováno jako otevřené rozhraní založené na jazyce Java. Podobně jako applety, což jsou programy napsané v jazyce Java, které se u klienta spouští v kontextu prohlížeče, jsou servlety programy napsané v jazyce Java, které se spouští na serveru v kontextu serveru. Aby se servlet mohl spustit potřebuje dvě podmínky:

- Na serveru musí být k dispozici Java Virtual Machine (JVM)
- WWW server musí podporovat tzv. Java Servlet API

První podmínka je prakticky splněna pro všechny současné moderní operační systémy. Druhá je splněna pro prakticky většinu dnešních významných a komerčně používaných serverů. Součástí vývojového balíku pro vývoj servletů (Java Servlet Development Kit) jsou ukázkové implementace Java Servlet API pro následující servery: *Netscape*, *Internet Information Server* a *Apache*. Tyto ukázkové implementace nejsou však plnohodnotné a určené pro rutinní profesionální provoz. Nicméně existují další implementace Java Servlet API, které jsou v současné době k dispozici.

- *JRun* firmy Live Software, který podporuje následující servery: MS IIS 3.0 a 4.0, MS Personal Web Server a O'Reilly's WebSite Professional. Ve vývoji je dále podpora pro Macintosh Web server WebStar
- *Apache Jserv*, je GNU projekt, původně podporující pouze server Apache. V současné době jsou zahájeny práce na podpoře NSAPI kompatibilních serverů (Netscape aj.) a ISAPI kompatibilních serverů (Microsoft, O'Reilly aj.)
- *ServletExpress*, firmy IBM, který podporuje následující produkty: Lotus Domino Go Webserver, Apache Server, Microsoft IIS a Netscape Enterprise Server.

Druhý ze zmíněných projektů, používá poněkud složitější implementaci Java Servlet API. *Apache Jserv* je samostatný proces, který na jedné straně komunikuje s částí rozhraní, která je přímo napojená na interní rozhraní daného serveru (Apache moduly, NSAPI, ISAPI) pomocí Apache Jserv Protokolu a na straně druhé obsahuje Java Servlet API pro spouštění servletů. Servlety tedy již neběží v kontextu WWW serveru.

Podobně jako u FastCGI zůstávají servlety v paměti a existuje možnost předávání informací mezi jednotlivými požadavky. Na rozdíl od FastCGI je výhodou větší rychlost, například oproti FastCGI aplikaci napsané například v Perlu, a jsou zachovány všechny výhody jazyka Java, jako je například bezpečnost nebo přenositelnost.

Zajímavá je možnost prostředku *LiveSite*. Jedná se o servlet, který funguje jako tzv. *filter*. Umožňuje tvorbu HTML stránek obsahující tzv. *makra*. Pokud klient o takovou stránku požádá, je předána servletu *LiveSite* ke zpracování, ten výše zmíněná makra vykoná (jedná se o makra umožňující práci se soubory, komunikaci s databází, atp.) a jejich výskyt v HTML dokumentu nahradí výsledky jejich činnosti. Jedná se defakto o totéž co je známo pod názvem Active Server Pages (ASP) od firmy Microsoft. *LiveSite* navíc umožňuje, aby uživatel definoval svá makra a přiřazoval jim své servlety. Uživatel má tedy v rukou prakticky neomezený rozvoj této koncepce.

Existuje také servlet, který dovoluje vykonávat přímo kód v jazyce Java, který je umístěn do stránek HTML, mezi značky <JAVA> a </JAVA>. Pochopitelně, vykonávání tohoto kódu se děje na straně serveru a značky včetně jimi ohraničeného obsahu jsou nahrazeny výstupem provedeného Java programu.

Jednoduchý zdrojový text Java servletu, který vrátí uživateli stránku s jeho jménem a počtem přístupů je uveden na příl. 6.

3.2. Proprietní rozhraní

Jedná se většinou o firemní rozhraní jednotlivých firem, které spolu zápasí na trhu WWW serverů. Jejich výhodou je větší výkon a často specializované funkce, které využívají výhod jednotlivých serverů. Nevýhodou je nutnost psát aplikace komunikující s těmito rozhraními většinou v jazyce C a poměrná složitost těchto rozhraní. Běžný vývojář WWW aplikací proto nepracuje přímo s těmito rozhraními, ale s prostředky vyšší úrovně, které jsou nad těmito rozhraními postaveny.

3.2.1. ISAPI (*Internet Server API*)

ISAPI od firmy Microsoft je určeno pro komunikaci s WWW servery firmy Microsoft. Aplikace, které používají toto rozhraní jsou kompilovány jako *dynamic-link libraries* (DLLs) a jsou serverem při jeho startu zavedeny do jeho kontextu. Výhodou ISAPI aplikací je pochopitelně jejich rychlost, nevýhodou je zejména složitý návrh a skutečnost, že špatně napsaná aplikace může celý WWW server vyřadit z činnosti.

ISAPI aplikace mohou pracovat ve dvou funkcích, jako *služba* (obr. č. 7) nebo jako *filter* (obr. č. 8). Vzhledem k tomu, že se nedá předpokládat, že běžný vývojář WWW aplikací bude psát své aplikace pro ISAPI, firma Microsoft připravila pro vývojáře nadstavby nad tímto rozhraním, které budou zmíněny v dalším textu. Zde je na místě pouze poznámka, že existuje trh s ISAPI aplikacemi (například URL) a jednou z nich je například interpreter jazyka Perl, realizovaný jako ISAPI aplikace od firmy Hip, Inc., který je určen pro převod pomalých CGI aplikací napsaných v tomto jazyce do prostředí ISAPI.

3.2.1.1. IDC (*Internet Database Connector*)

Je poměrně jednoduchá nadstavba nad rozhraním ISAPI (pracuje jako ISAPI filtr) od firmy Microsoft. Princip její činnosti je znázorněn na obr. 9. Základem IDC aplikace jsou dva soubory:

- .IDC soubor, který v sobě obsahuje informaci, s jakou databází a kdo se má spojit a jaký soubor šablony (.HTX) se má použít pro výstup. Také obsahuje vlastní dotaz do databáze. V tomto souboru mohou být formální parametry, které se při zpracování .IDC souboru nahradí skutečnými. Ukázka jednoduchého .IDC souboru je na příl. 7.

- .HTX soubor. Jedná se šablonu HTML stránky. V tomto souboru je možné vkládat mezi speciální formátovací znaky (%) jednoduché příkazy, které jsou nahrazeny výsledky dotazu do databáze nebo jinými údaji. Je také možné využití jednoduché řídicí logiky (podobné jednoduchému programovacímu jazyku). Ukázka takové šablony na příl. 8. Výsledná stránka (generovaná na straně serveru podle zde uvedených souborů .IDC a .HTX), jak je zobrazena uživateli, je na obr. 10

Přes poměrně velké možnosti IDC není tato nadstavba firmou Microsoft dále rozvíjena a byla nahrazena daleko komplexnější koncepcí *Active Server Pages*.

3.2.1.2. ASP (*Active Server Pages*)

Active Server Pages (dále jen ASP) je nadstavba nad ISAPI rozhraním (pracuje jako filtr) firmy Microsoft, využívající jako programovacího prostředku uvnitř HTML šablon jazyků VBscript a Jscript a dále využívající možnosti Active-X komponent.

Základní princip ASP je znázorněn na obr. 11. Stránky HTML se kterými ASP pracuje v sobě obsahují kód skriptovacího jazyka a mají standardní koncovku .asp. Při požadavku klienta na takovou stránku je tato stránka předána ke zpracování komponentě ASP. Tato komponenta nechá vykonat ve stránce obsažené části programovacího kódu, které nahradí jejich výsledkem (výsledkem může být třeba výstup z databáze, pokud se programovací kód spojí s databází a získá z ní výsledky na svůj dotaz) a výslednou stránku vrátí WWW serveru, který ji posléze zašle klientu.

Ukázka takového jednoduchého do HTML stránky vloženého kódu, který komunikuje s databází je na příl. 10. (užití VBscriptu) a na příl. 11 (užití JScriptu). Výsledná vygenerovaná stránka je na obr. 12. Jak je vidět ze zdrojového textu, komunikace se děje prostřednictvím Active-X komponent. Zabudovaný skriptovací jazyk totiž dokáže pouze řídit činnost programu, pracovat s proměnnými a provádět výstup do HTML stránky. Veškeré další činnosti se dějí prostřednictvím Active-X komponent, které jsou zde defakto použity na místo dynamických knihoven. Standardní součástí ASP jsou následující Active-X komponenty:

- *Active-X Data Objects* (ADO), je komponenta určená pro přístup k databázím, prostřednictvím rozhraní ODBC. Je to pro vývojáře IS nejdůležitější komponenta standardně dodávaná s ASP.
- *Ad Rotator*, je komponenta umožňující při každém přístupu na ASP stránku zobrazit jiný reklamní obrázek
- *Browser Capabilities*, je komponenta, která vývojáři stránek ASP zpřístupní informace o klientem použitým prohlížeči.

- *Content Linking*, je komponenta, která generuje odkazy na stránky na základě textového souboru.
- *File Access Component*, je komponenta, která vývojáři zpřístupní práci se soubory na straně serveru.

Nejdůležitější pro tvůrce IS aplikací je bezpochyby komponenta ADO. Pomocí této komponenty může komunikovat prostřednictvím rozhraní ODBC s lokálními i vzdálenými databázemi. Nejjednodušší, i když určitým způsobem omezená, je možnost využití ODBC driverů k databázím typu .mdb (*Microsoft Access Database*).

V současné době vznikl na internetu velice bohatý a rychle se rozvíjející se trh s těmito komponentami. Tyto Active-X komponenty je možné vyvíjet v mnoha programovacích jazycích, například C++, Visual Basic, Delphi a dalších, tak jak tomu bylo uvedeno v části věnované Active-X objektům na straně klienta.

Vzhledem k tomu, že všechny tyto komponenty mají prostřednictvím ASP poměrně blízkou vazbu na WWW server, hrozí ovšem nebezpečí, že špatně napsaná komponenta může server zastavit.

Velice zajímavá je možnost tzv. *Active-X Scripting Engines*. Koncepce ASP je totiž velice modulární. Samotné interpretery skriptovacích jazyků jsou totiž ve formě Active-X komponent. Přestože Microsoft dodává s ASP pouze dvě Active-X Scripting Engines a to pro jazyky VBscript a JScript, existuje pro vývojáře možnost vytvořit a do ASP zabudovat další téměř libovolnou Active-X Scripting Engine. Očekává se vznik Active-X Scripting Engines pro jazyky Perl, Python a Tk/Tcl.

V současné době je ASP velice populárním a jedním z nejrozšířenějších prostředků pro tvorbu dynamický HTML stránek a aplikací IS na Webu. Tvorba ASP stránek je podporována produkty Microsoftu, jako je například FrontPage (HTML editor a správce WWW serverů) a Visual InterDev (vývojové prostředí pro tvorbu dynamických intranetových aplikací). Nicméně vzhledem ke koncepci – skriptovací jazyk vložený do HTML stránek, nejsou zatím vizuální prostředky pro návrh dynamických stránek tak dobré, jak by si vývojář přál.

3.2.2. Apache moduly

Nejrozšířenější WWW server Apache nemá pro své interní rozhraní nijaké zvláštní jméno. Hovoří se prostě o modulech. Rozhraní je definováno pro jazyk C. Modul vytvořený v jazyce C je nutné přeložit a celý server, který je ostatně jako každý GNU program k dispozici ve zdrojovém tvaru, znovu vygenerovat. Rozhraní modulů serveru Apache poskytuje sice poměrně velké možnosti ohledně spolupráce s WWW serverem (modul může defakto Apache server velice bohatě rozšiřovat skoro ve všech jeho interních činnostech, například ověření uživatele, logování požadavků, filtrování stránek, kódování češtiny a mnoha

dalších), nicméně rozhraní je dle mých zkušeností poněkud málo dokumentováno a pro běžného uživatele je příliš komplikované.

Podobně jako u ISAPI rozhraní, existují pro běžné uživatele nadstavby se kterými může vývojář snadněji pracovat.

3.2.2.1. PHP/FI

PHP/FI je nadstavba určená pro tvorbu dynamických stránek na straně serveru zejména pro server Apache. Její funkce je velice podobná činnosti ASP. Šablony HTML stránek obsahují programy ve skriptovacím jazyce podobném jazyku Javascript. Modul PHP/FI tyto části programu vykonává a kód programu nahradí jeho výsledkem. V současné době existují PHP/FI verze 2.0 a nová verze 3.0, které se od sebe poněkud odlišují na úkor přenositelnosti aplikací.

Velkou výhodou PHP/FI je skutečnost, že jazyk obsahuje příkazy pro práci s databázemi, které jsou k dispozici pro systém Linux (a jiné zejména UNIXové systémy). Jedná se o databáze: Adabas, dBase, filePro, mSQL, MySQL, iODBC (freeODBC), ORACLE, Postgres95/PostgresSQL, Solid, Sybase, Sybase-CT, Velocis, Unified ODBC a LDAP. Jazyk PHP/FI je objektově orientován a poskytuje bohatou sadu příkazů. Poskytuje také API pro jazyk C, což může být využito pro jeho další rozšiřování. Ukázka jednoduchého kódu v jazyce PHP/FI je na příl. 9

Přestože je PHP/FI určeno především pro server Apache, může být spuštěno také jako CGI aplikace na serverech Netscape a NCSA. V oblasti Linuxu se rozhodně jedná o nejpoblárnější systém tohoto typu.

3.2.2.2. mod_isapi

Jedná se o poměrně atypickou záležitost. Protože server Apache je v současné době možné provozovat také na platformě Windows NT a nahradit jím MS IIS. Vznikl požadavek na možnost využití ISAPI aplikací napsaných pro MS IIS s Apache serverem. Výsledkem byl vznik Apache modulu, který umožňuje aby ISAPI aplikace (bohužel nikoliv aplikace typu filtr a tedy ani samotné ASP) bylo možné provozovat s Apache serverem. Ani u ostatních ISAPI aplikací není zaručena naprostá kompatibilita a schopnost jejich provozu.

3.2.3. NSAPI (Netscape Server API)

Pochopitelně i největší konkurent Microsoftu na poli Internetu firma Netscape, umožňuje psát vývojářům aplikace rozšiřující možnosti jejich WWW serverů. Rozhraní má název NSAPI a poskytuje programátorům podobné možnosti jako rozhraní ISAPI firmy Microsoft.

3.2.3.1. LiveWire (Server Side JavaScript)

Podobně jako firma Microsoft s jejím systémem ASP, umožňuje firma Netscape programování dynamických HTML stránek generovaných na straně serveru pomocí produktu s názvem *LiveWire*. Základem pro programování

dynamických stránek na straně serveru je pochopitelně JavaScript, jazyk vytvořený právě firmou Netscape.

3.3. Další WWW servery a jejich rozhraní

Mimo nejrozšířenějších WWW serverů (Apache, Netscape, IIS) existuje celá řada dalších serverů, které poskytují nejrůznější možnosti pro tvorbu IS aplikací. Není možné se tímto příspěvkem o všech zmínit. Některé z nich (Java Web Server) již byly zmíněny v předchozím textu. Zde jen velice stručně:

- *Novell* Původně vlastní WWW server, umožňující spouštět CGI skripty v jazyce Perl a Netbasic na Netware serveru. Také možnost spouštět CGI skripty na vzdáleném počítači. Nyní možnost spouštění aplikací v jazyce Java na Netware serveru. V současné době vlastní server od firmy Novell nahrazen serverem od firmy Netscape.
- *IBM* Firma IBM jako WWW server pro své produkty (např. operační systém OS/2) používá produkt *Lotus Domino Go Webserver*. Tento WWW server podporuje pochopitelně CGI skripty, vlastní interní rozhraní s názvem GWAPI určené pro programátory v jazyce C a navíc podporuje Java servlety, pomocí vlastního produktu *ServletExpress*.

4. Závěr

Problematika intranetových aplikací je téma velice rozsáhlé, a proto není tento příspěvek rozhodně vyčerpávající. Jeden z důvodů byla skutečnost, že některé aplikace osobně provozuji a jiné znám pouze z jejich dokumentace a dalších informačních zdrojů. Další důvody byly zejména časové, protože se jedná o rozsáhlou problematiku, která se navíc neustále rozšiřuje. Nicméně jsem se snažil v rámci svých možností se v tomto příspěvku zmínit o všech významných serverech, rozhraních a programovacích jazycích, které se k dané problematice vztahují a začlenit je do kontextu ve kterém se vyskytují.

Doufám, že čtenáři tento příspěvek poskytne aspoň částečný, když ne ucelený, přehled této zajímavé a v blízké budoucnosti (nebo dokonce již přítomnosti) nesmírně důležité problematiky. Měl jsem sám možnost si ověřit prakticky tvorbu komerční intranetové aplikace technologii ASP (obr. 13). Na základě takto získaných zkušeností jsem si uvědomil, že vytváření IS aplikací pro intranet má sice mnoho společných prvků s klasickou tvorbou IS aplikací, ale též mnohé odlišnosti, například zatím poněkud slabé možnosti ladění a vizuálního návrhu, nutnost jiného členění kódu aplikace s ohledem na využití klienta a serveru a mnohé další rozdíly.

Celosvětová komunikační *pavučina*, čili Web je realita a stále více firem do ní přesouvá své komerční aktivity. Výhody zmíněné v úvodu tohoto příspěvku jsou očividné. Proto je nutné s tímto globálním trendem zachovat krok a problematice Internetu a intranc.u se cíleně věnovat. Je to nutnost pro každého moderního programátora, informatika i informačního manažera!

Literatura

- [1] FLANAGAN, David: *Java in a Nutshell*, O'Reilly & Associates, Inc., 1996, U.S.A.,
- [2] WALL LARRY, CHRISTIANSEN TOM, SCHWARTZ L. RANDAL: *Programování v jazyce PERL*, Computer Press, 1997, Praha
- [3] SATRAPA PAVEL, *World-Wide Web*, Neokortex, 1996, Praha
- [4] HLAVENKA, J., SEDLÁŘ, R., HOLČÍK, T., ŠEBESTA, M., BOTÍK, R., *Vytváříme WWW stránky*, Computer press, 1997
- [5] MÍKLE PAVEL, *DHTML, referenční příručka*, UNIS Publishing, 1997, Brno
- [6] PLECHÁČ VLADIMÍR: *Microsoft VisualDev*, Gcomp. 1997, Praha

URL reference

W3C - The World Wide Web Consortium (Informace o WWW, HTML, atp.) HTTP://WWW.W3.ORG/
W3C (Specifikace HTML 4.0) http://www.w3.org/TR/REC-html40/
JavaSoft (Java Applets) http://www.javasoft.com/applets/index.html
MICROSOFT (ACTIVE-X) http://www.microsoft.com/intdev/sdk/
MICROSOFT (DYNAMIC HTML) http://www.microsoft.com/workshop/author/dhtml/
NETSCAPE (DYNAMIC HTML) http://developer.netscape.com/docs/manuals/dynhtml.html
NETSCAPE (JAVASCRIPT) http://developer.netscape.com/docs/manuals/javascript.html
MICROSOFT (JSCRIPT) http://premium.microsoft.com/msdn/library/tools/JScript/html/jstutor.htm
MICROSOFT (VBSCRIPT) http://premium.microsoft.com/msdn/library/tools/vbscript/html/vbstutor.htm
CGI http://agora.leeds.ac.uk/nik/Cgi/start.html
FASTCGI http://www.fastcgi.com/
JAVASOFT (JAVA SERVLETS) http://www.javasoft.com/products/java-server/servlets/index.html
JAVASOFT (JAVA WEB SERVER) http://jserv.javasoft.com/products/webserver/index.html
MICROSOFT (ISAPI) http://www.microsoft.com/win32dev/apiext/isalegal.htm
MICROSOFT (IDC/HTX) http://premium.microsoft.com/msdn/library/periodic/d1/sa1096.htm
MICROSOFT (ACTIVE SERVER PAGES) http://www.microsoft.com/iis/guide/aspooverview.asp
APACHE SERVER (MODULES) http://www.apache.org/
PHP/FI http://www.php.net/
NETSCAPE (NSAPI) http://developer.netscape.com/docs/manuals/enterprise/bookshelf/index.htm
NETSCAPE (LIVEWIRE) http://developer.netscape.com/docs/manuals/enterprise/wrijsap/index.htm

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@fei.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

Příloha

K313
Katedra elektrotechnologie

Seznam pracovníků K313

Foto	Jméno	Pracovní číslo	Učinnost	Titul
	Doc. Ing. Pavel Blahůš CSc.	34352122	vedoucí katedry	vedoucí katedry
	Romana Havránková	2123	Sekretářka	sekretářka
	Doc. Ing. Jiří Štěrba CSc.	2179	Zástupce vedoucí katedry	zástupce vedoucí katedry
	Doc. Ing. Jan Urbánek CSc.		Typograf katedry	typograf katedry

obr. 1 Stránka HTML s tabulkou a obrázky generovaná ASP

Uprava předmětu

Název:

Kód:

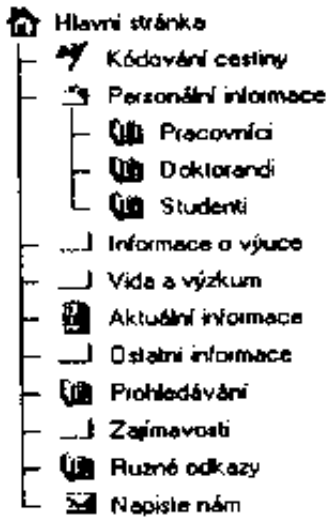
Semestr:

Počet kreditů:

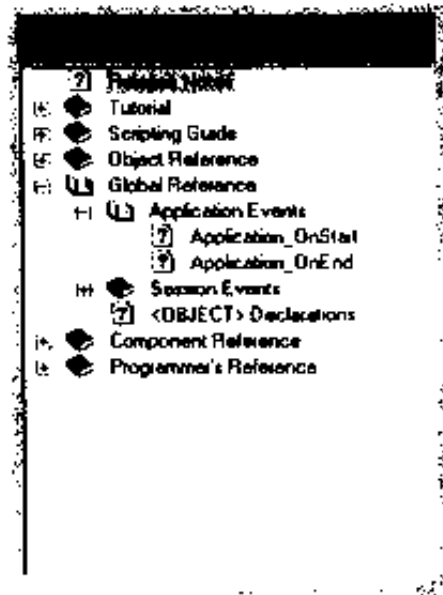
ANALÝZA A SYNTÉZA DYNAMICKÝCH SYSTÉMŮ

1. Úvod - úvodní pojmy a významnosti, dynamický systém
2. Základní metody řešení, dynamická, lineární, triviální řešení
3. Vnější vlivy dynamického systému a jeho řešení
4. Vnitřní vlivy a vnitřní přechod, řízení, řízení s zpětnou vazbou
5. Modelování a řízení, příklady - cíle a metody
6. Stabilita řešení dynamického systému, kritéria stability
7. Programové podpory pro počítačové řešení, výpočet
8. Reprezentace matematických systémů, příklady, řešení

obr. 2 Stránka HTML s formulářem generovaná ASP



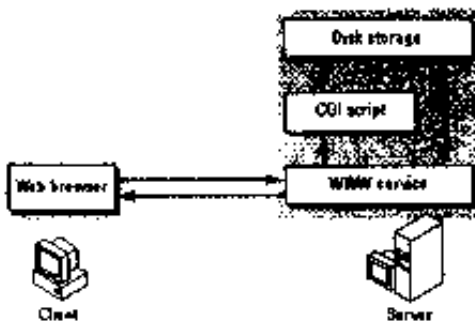
obr. 3 Ukázka Java appletu



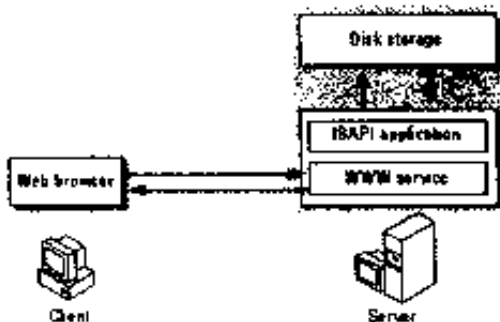
obr. 4 Ukázka Active-X objektu



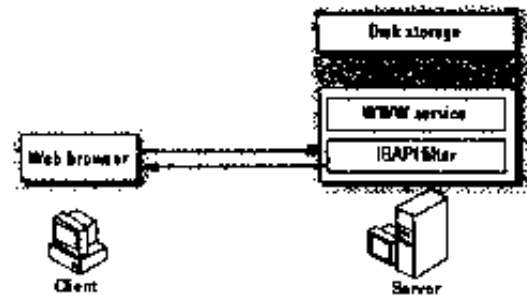
obr. 5 Ukázka tlačítka, které generuje události obsloužené metodami v jazyce Javascript nebo Vbscript



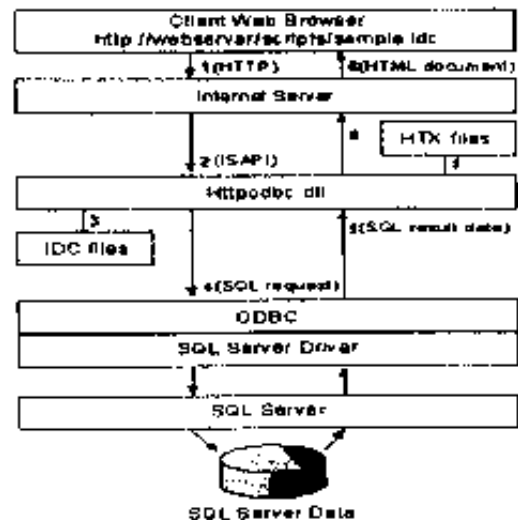
obr. 6 Princip funkce CGI



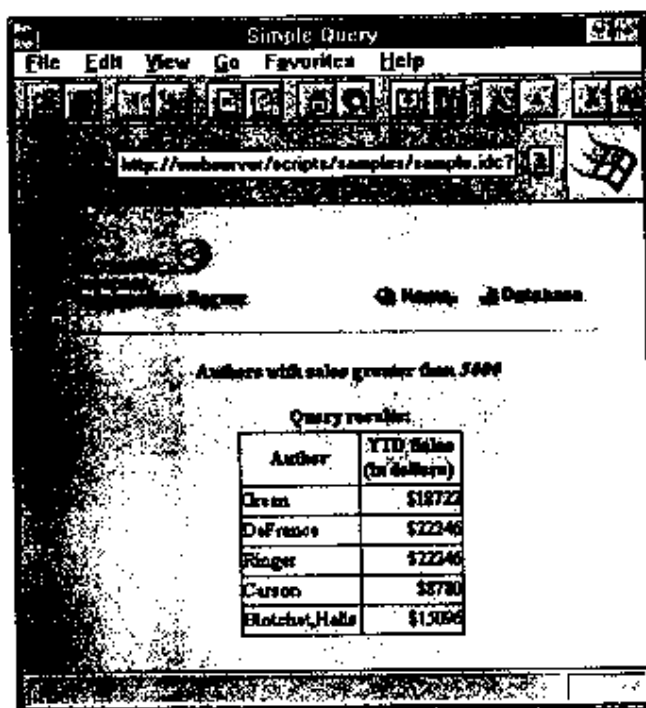
obr. 7 ISAPI jako služba



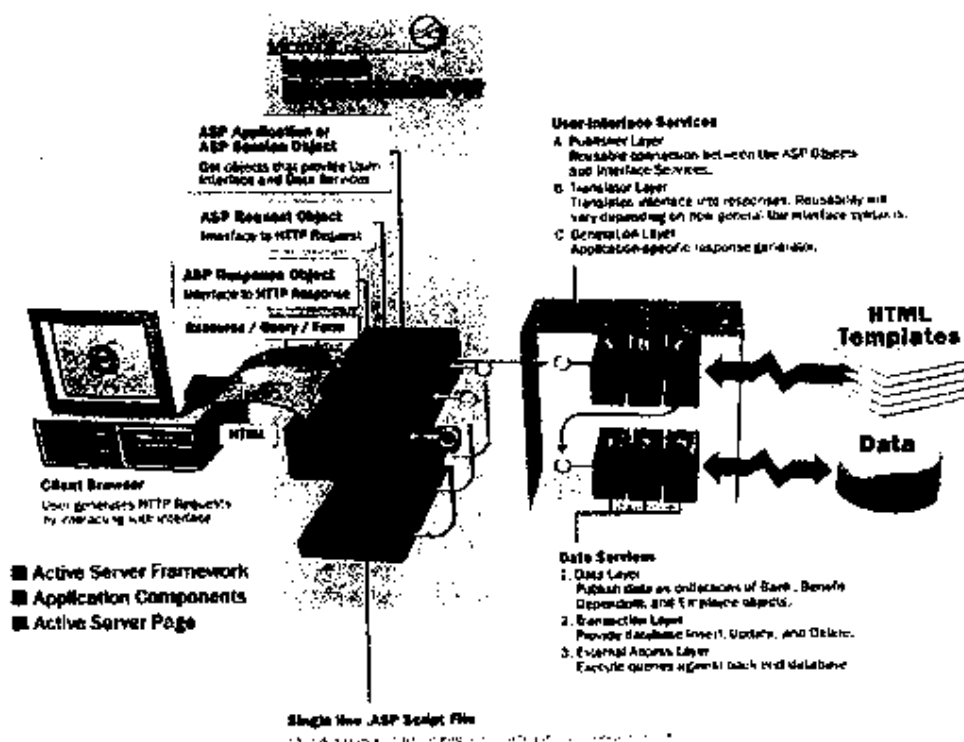
obr. 8 ISAPI jako filtr



obr. 9 Princip IDC



obr. 10 Ukázka stránky generované pomocí IDC/HTX



obr. 11 Princip ASP (Active Server Pages)

ActiveX Data Objects (ADO)

OrderID	CustomerID	EmployeeID	OrderDate	ProductID	ProductName	ShipCountry	ShipCity	ShipAddress	ShipCity	ShipStateOrProvince	ShipPostal
1	1	5	22.1995	52	Java	Yorba	Yorba	Let's Stop N Shop	87 Park St Suite 3	San Francisco	CA 94117
2	1	5	14.1.1995	52	Java	Yorba	Yorba	Let's Stop N Shop	87 Park St Suite 3	San Francisco	CA 94117
3	1	1	12.4.1995	81	Java	Yorba	Yorba	Let's Stop N Shop	87 Park St Suite 3	San Francisco	CA 94117
4	1	2	21.3.1995	125	Java	Yorba	Yorba	Let's Stop N Shop	87 Park St Suite 3	San Francisco	CA 94117
5	1	1	23.6.1995	140	Java	Yorba	Yorba	Let's Stop N Shop	87 Park St Suite 3	San Francisco	CA 94117
6	2	3	1.2.1995	122	Java	Phillips	Phillips	Old World Delicatessen St	2740 Bering	Anchorage	AK 99508
7	2	5	1.4.1995	353	Java	Phillips	Phillips	Old World Delicatessen St	2740 Bering	Anchorage	AK 99508
8	2	1	9.4.1995	446	Java	Phillips	Phillips	Old World Delicatessen St	2740 Bering	Anchorage	AK 99508
9	2	2	11.2.1995	333	Java	Phillips	Phillips	Old World Delicatessen St	2740 Bering	Anchorage	AK 99508
10	2	1	12.6.1995	600	Java	Phillips	Phillips	Old World Delicatessen St	2740 Bering	Anchorage	AK 99508
11	3	2	14.2.1995	55	Perl	Wilson	Wilson	Rattlesnake Canyon Quarry	2817 Milton Dr	Albuquerque	NM 87110
12	3	4	25.2.1995	124	Perl	Wilson	Wilson	Rattlesnake Canyon Quarry	2817 Milton Dr	Albuquerque	NM 87110
13	3	2	26.2.1995	222	Perl	Wilson	Wilson	Rattlesnake Canyon Quarry	2817 Milton Dr	Albuquerque	NM 87110
14	3	4	15.4.1995	555	Perl	Wilson	Wilson	Rattlesnake Canyon Quarry	2817 Milton Dr	Albuquerque	NM 87110

obr. 12 Stránka ASP jako výsledek práce s objektem ADO

Armáda nestvůr
Sci-Fi

Autor:
Darlton, Clark - autor
Kučák, Jaroslav - překladatel

Vydavatelství:
Edice: Království Perry Rhodan

Cena: 21 Kč

ISBN: 80-740-0177-3
Rok vydání: 1978
Rok: 2007

Klíčová slova: sci-fi, fantasy, horor

Vstříkněte kofíček:

Stručně...
Aster, Rhodan a Felmer jsou dopraveni na centrální planetu Pořbů. Při ohrožení vlastních životů se jim podaří deaktivovat tu část programu u Pořbů, která vyvolává jejich zrod v obřích organickému životu.

obr. 13 Ukázka komerční aplikace založené na technologii ASP

```

<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1250">
<title></title>
</head>

<body>
<p><applet code="navigator.class" codebase="."
align="baseline" width="200" height="1000" name="navigator">
<param name="autosizing" value="true">
<param name="line_style" value="solid">
<param name="image_dir" value="./images/">
<param name="image0" value="zip,zip.gif,klikni...">
<param name="image1" value="flag,flag.gif,klikni...">
<param name="image2" value="doc,doc.gif,klikni...">
<param name="image3" value="globe,globe.gif,klikni...">
<param name="image4" value="home,home.gif,klikni...">
<param name="image5" value="mail,mail.gif,klikni...">
<param name="item0" value="0,home,hlavni,/hlavni.asp,Hlavní stránka,úvodní stránka">
<param name="item1" value="1,Flag,hlavni,/asp/codepage.asp,Kódování cestiny,zmína kódování
ee7tiny">
<param name="item2" value="1,closed,hlavni,/osobni/default.asp,Personální informace,informace
o pracovnících a studentech">
<param name="item3" value="2,url,hlavni,/asp/pracovnici.asp,Pracovníci, pracovníci katedry
elektrotechnologie">
<param name="item4" value="2,url,hlavni,/asp/doktorandi.asp,Doktorandi">
<param name="item5" value="2,url,hlavni,/asp/studenti.asp,Studenti">
<param name="item6" value="1,closed,hlavni,/vyuka/default.asp,Informace o výuce">
--- Zde je vynecháno několik řádek ---
<param name="item27" value="1,closed,hlavni,/links/default.htm,Ruzné odkazy,ruzné zajímavé
odkazy na jiné WWW servery">
<param name="item28" value="1,mail,hlavni,/maillist.asp,Napiste nám,">
</applet></p>
</body>
</html>

```

přil. 1 Ukázka Java appletu vloženého do HTML stránky

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

<HTML>
<HEAD>
<TITLE>Contents</TITLE>
</HEAD>

<body bgcolor="#C0C0C0" TOPMARGIN=0 LEFTMARGIN=0>
<br>
<ul>
<table border="0" cellpadding="0" cellspacing="0" width="236">
<tr>
<td valign="top" colspan="4" bgcolor="#3366CC">

</td></tr>
<tr>
<td valign="top" colspan="4" bgcolor="#CCCCCC">
<OBJECT id=hhctrl type="application/x-oleobject"
classid="clsid:adb880a6-d0ff-11cf-9377-00aa003b7a11"
codebase="hhctrl.cab#version=4,0,0,31"
width=236
height=315
>
<PARAM name="Command" value="Contents">
<PARAM name="Item1" value="hhctrl.hhc">
</OBJECT>
</td></tr>
</table>
</ul>
</BODY>
</HTML>

```

přil. 2 Ukázka Active-X objektu vloženého do HTML stránky

JScript	VBScript
<pre> <SCRIPT LANGUAGE="JavaScript"> <!-- function Press_OnClick () { alert("It works!") } --> </SCRIPT> </HEAD> <P> <H3>Hello JScript Sample </H3> <FORM Name="form1"> <INPUT TYPE="BUTTON" VALUE="Click me" NAME="BtnHello" OnClick="Press_OnClick()"> </FORM> </pre>	<pre> <SCRIPT LANGUAGE="VBScript"> <!-- Sub Press_OnClick () alert("It works!") End Sub --> </SCRIPT> </HEAD> <P> <H3>Hello VBScript Sample</H3> <FORM Name="form1"> <INPUT TYPE="BUTTON" VALUE="Click me" NAME="BtnHello" OnClick="Press_OnClick()"> </FORM> </pre>

přil. 3 Obsluha události generované tlačítkem v jazyce Javascript (JScript) a VBscript

```

#!/usr/local/bin/perl
# Tiskneme hlavičku
print("Content-type: text/html\r\n\r\n")
# Tiskneme uvítání
print("<h1>CGI Hello! (Perl)</h1>\n")
# Tiskneme identifikaci počítače
print(" running on host <i>$_ENV{'SERVER_NAME'}</i>")
# Konec scriptu
return (0);

```

přil. 4 Jednoduchý CGI script

```

#!/usr/local/bin/perl
# Musíme použít Fast CGI knihovnu
use FCGI;
# Nastavíme čítač přístupů
$count = 0;
# Hlavní smyčka scriptu
while(FCGI::accept() >= 0) {
    # Tiskneme hlavičku
    print("Content-type: text/html\r\n\r\n")
    # Tiskneme uvítání
    print("<h1>FastCGI Hello! (Perl)</h1>\n")
    # Tiskneme počet přístupů
    print("Request number ", $++count)
    # tiskneme identifikaci počítače
    print(" running on host <i>$_ENV{'SERVER_NAME'}</i>")
}
# Konec scriptu
return (0);

```

přil. 5 Jednoduchý FastCGI script

```

import java.io.*; import javax.servlet.*; import javax.servlet.http.*;
public class firstServlet extends HttpServlet {

    public void init(ServletConfig conf) throws ServletException {
        super.init(conf);
        connections = 0;
    }

    public void service(HttpServletRequest req, HttpServletResponse res) throws IOException{
        //Get a channel to the Web browser, so we can send output
        ServletOutputStream out = res.getOutputStream();
        res.setContentType("text/html"); // Required for HTTP
        out.println("<HTML><HEAD><TITLE>Servlets 101</TITLE></HEAD>");
        out.print("Say hello to Java Servlet Programming. ");
        out.println(req.getParameter("userName"));
        out.print("<p>You are caller number ");
        connections++;
        out.println(Integer.toString(connections));
        out.close(); //Close the output stream
    }

    int connections;
}

```

přil. 6 Jednoduchý Servlet

```

DataSource: web sql
Username: sa
Template: sample.htx
SQLStatement:
+SELECT au_lname, ytd_sales
+from pubs.dbo.titleview
+where ytd_sales>5000

```

přil. 7 Soubor IDC

```

<HTML>
<HEAD><TITLE>Authors and YTD Sales</TITLE></HEAD>
<BODY BACKGROUND="/samples/images/backgrnd.gif">
<BODY BGCOLOR="FFFFFF">
<TABLE>
<TR><TD><IMG SRC="/samples/images/SPACE.gif" ALIGN="top" ALT=" "></TD>
<TD><A HREF="/samples/IMAGES/db_mh.map"><IMG SRC="/SAMPLES/images/db_mh.gif" ismap BORDER=0
ALIGN="top" ALT=" "></A></TD>
</TR><tr><td></td><td><hr>
<font size=2>
<CENTER>
<?if idc.sales eq "">
<H2>Authors with sales greater than <I>5000</I></H2>
<?else>
<H2>Authors with sales greater than <I><?idc.sales?</I></H2>
<?endif?><P>
<TABLE BORDER=
<?legendetail?>
<?if CurrentRecord EQ 0 ?>
<caption>Query results:</caption>
<TR><TH><B>Author</B></TH><TH><B>YTD Sales<BR>(in dollars)</B></TH></TR>
<?endif?>
<TR><TD><?au_lname?</TD><TD align="right"><?ytd_sales?</TD></TR>
<?enddetail?>
<P></TABLE>
</center><P>
<?if CurrentRecord EQ 0 ?>
<I><B>Sorry, no authors had YTD sales greater than </I><?idc.sales?</B>
<P>
<?else?>
<HR><I>
The web page you see here was created by merging the results
of the SQL query with the template file SAMPLE.HTX.
<P>
The merge was done by the Microsoft Internet Database Connector and
the results were returned to this web browser by the Microsoft Internet Information Server.
</I>
<?endif?>
</font>
</td></tr></table>
</BODY></HTML>

```

přil. 8 Soubor HTX

```

<?php
if(!$_PHP_AUTH_USER) {
    Header("WWW-authenticate: basic realm=\"My Realm\"");
    Header("HTTP/1.0 401 Unauthorized");
    echo "Text to send if user hits Cancel button\n";
    exit;
} else {
    echo "Hello $_PHP_AUTH_USER.<P>";
    echo "You entered $_PHP_AUTH_PW as your password.<P>";
}
?>

```

přil. 9 Autentizace pomocí PHP/FI


```

<HTML>
<HEAD>
<TITLE>ActiveX Data Object (ADO)</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF>
<H3>ActiveX Data Object (ADO)</H3>
<%
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "ADOSamples"
Set RS = Conn.Execute("SELECT * FROM Orders")
%>
<P>
<TABLE BORDER=1>
<TR>
<% For i = 0 to RS.Fields.Count - 1 %>
<TD><B><% = RS(i).Name %></B></TD>
<% Next %>
</TR>
<% Do While Not RS.EOF %>
<TR>
<% For i = 0 to RS.Fields.Count - 1 %>
<TD VALIGN=TOP><% = RS(i) %></TD>
<% Next %>
</TR>
<%
RS.MoveNext
Loop
RS.Close
Conn.Close
%>
</TABLE>
<BR>
<BR>
<!--#include virtual="/ASPSamp/Samples/srcform.inc"-->
</BODY>
</HTML>

```

přil. 10 Stránka ASP, jazyk VBScript, užití objektu ADO

```

<%@ Language=JScript %>
<HTML>
<HEAD><TITLE>JScript and the ActiveX Data Object (ADO)</TITLE></HEAD>
<BODY BGCOLOR=#FFFFFF>
<H3>ActiveX Data Object (ADO)</H3>
<%
Conn = Server.CreateObject("ADODB.Connection")
Conn.Open("ADOSamples")
RS = Conn.Execute("SELECT * FROM Orders")
%>
<P>
<TABLE BORDER=1>
<TR>
<% for (i = 0; i < RS.Fields.Count - 1; i++) { %>
<TD><B><% = RS.Fields(i).Name %></B></TD>
<% } %>
</TR>

<% while (!RS.EOF) { %>
<TR>
<% for (i = 0; i < RS.Fields.Count - 1; i++) { %>
<TD VALIGN=TOP><% = RS(i) %></TD>
<% } %>
</TR>
<%
RS.MoveNext()
}
RS.Close()
Conn.Close()
%>
</TABLE>
<BR>
<BR>
<!--#include virtual="/ASPSamp/Samples/srcform.inc"-->
</BODY>
</HTML>

```

přil. 11 Stránka ASP, jazyk JScript, užití objektu ADO