

**UNIVERZITA PARDUBICE
FAKULTA EKONOMICKO–SPRÁVNÍ
KATEDRA INFORMAČNÍCH SYSTÉMŮ**

**PREZENTACE DAT Z FORMÁTU MS OFFICE
DO PROSTŘEDÍ WWW S POUŽITÍM TECHNOLOGIÍ ASP A ADO**

**BAKALÁŘSKÁ PRÁCE
2000**

**AUTOR: Tomáš Myšík
VEDOUCÍ PRÁCE: Ing. Jiří Petráš, CSc.**

**UNIVERSITY OF PARDUBICE
FACULTY OF ECONOMICS AND ADMINISTRATION
DEPARTMENT OF INFORMATION SYSTEM**

**THE PRESENTATION OF DATA OF MS OFFICE FORMAT INTO
WWW USING THE TECHNOLOGIES OF ASP AND ADO**

BACHELOR WORK

2000

**AUTHOR: Tomáš Myšík
SUPERVISOR: Ing. Jiří Petráš, CSc.**

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a že veškerou literaturu, kterou jsem použil, jsem uvedl v seznamu použité literatury.

28. 4. 2000

Tomáš Myšík

Poděkování

Tímto bych rád poděkoval za připomínky a odborné rady vedoucímu práce
Ing. Jiřímu Petrášovi, CSc.

SOUHRN

Autor: Tomáš Myšík

Název práce: Prezentace dat z formátu MS Office do prostředí WWW s použitím technologií ASP a ADO

Vedoucí práce: Ing. Jiří Petráš, CSc.

Tato práce se zabývá stručným teoretickým popisem technologií firmy Microsoft Active Server Pages (ASP) a ActiveX Data Objects (ADO), mimo jiné konkrétně popisuje vytvoření funkčního systému WWW stránek, jehož hlavním úkolem je zobrazovat pouze ta data z databáze MS Excel, která si uživatel pomocí internetového formuláře vyžádá.

Tato práce obsahuje 37 stran.

SUMMARY

Author: Tomáš Myšík

Title: The presentation of data of MS Office format into WWW using the technologies of ASP and ADO

Supervisor: Ing. Jiří Petráš, CSc.

This document is concerning the brief theoretical description of the firm Microsoft Active Server Pages (ASP) and ActiveX Data Objects (ADO). Among other things it describes the creation of the functional system of WWW pages, whose main taste is to show only that kind of data from the database MS Excel, which the user can claim using the Internet form.

This document contains 37 pages.

OBSAH:

I. Úvod	3
II. Cíle	4
III. Teoretická část	5
1. Internet	5
1.1. Statický Internet	6
1.2. Dynamický Internet	7
1.2.1. Skriptování na straně klienta	7
1.2.2. Skriptování na straně serveru	8
1.2.2.1. Common Gateway Interface	8
1.2.2.2. Microsoft Active Server Pages	9
2. Vývoj ASP a ADO	12
3. Alternativy k MS ASP	13
4. Technologie Microsoft Active Server Pages	13
4.1. Objekt Application	15
4.2. Objekt Session	15
4.3. ObjektObjectContext	16
4.4. Objekt Server	16
4.5. Objekt Response	16
4.6. Objekt Request	17
4.7. Komponenty ASP	19
IV. Praktická část	21
1. Co bylo úkolem	21
2. Potřebné součásti	21
3. Schéma WWW stránek	23
4. Tvorba WWW stránek	23
4.1. Karosa.xls	24
4.2. Styly.css	24
4.3. Skripty.js	25
4.4. Formular.htm	26
4.4.1. Formular.htm – 1. verze	27

4.4.2. Formular.htm – 2. verze	28
4.4.3. Formular.htm – 3. verze	29
4.5 Karosa.asp	30
4.6. Typy.asp	33
4.7. Typy2.asp	33
V. Závěr	36
VI. Seznam použité literatury	37
VII. Přílohy	38

SEZNAM OBRÁZKŮ:

Model požadavek – odpověď	6
Princip fungování CGI	9
Princip fungování ASP	10
Schéma WWW stránek	23
Internetový formulář	26
Formulář se skrytými poli	29
Zobrazení výsledků	33

SEZNAM TABULEK:

Výhody a nevýhody „inteligentních“ editorů	22
Karosa.xls – ukázka	24

I. ÚVOD

Internet. Tento termín se dnes skloňuje snad ve všech pádech, neboť se jedná oblast, která v posledních letech prožívá bouřlivý rozvoj. Ještě před deseti lety znal toto slovo jen málokdo – v podstatě pouze pár počítačových šílenců a studentů škol, které se zabývají informatikou. Na obrovském nárůstu popularity Internetu má největší podíl nepochybně služba World Wide Web, která do té doby „vědeckému“ Internetu vtiskla jakousi lidštější tvář.¹ Postupnou přeměnu tváře a využití Internetu je velice dobře možné vyzorovat z chování firem – stačí srovnat inzertní část tisku zhruba za posledních 5 let. Zpočátku se pouze zřídka objevovaly e-mailové adresy firem, a to ještě převážně těch, které se počítači zabývají. S postupem času se ale e-mailové adresy začaly objevovat mnohem více, přičemž později se začínají objevovat i adresy webových prezentací. V dnešních novinách byste naopak velice těžko hledali firmu, která neposkytuje informace o svých produktech nebo službách také na Internetu. Dokonce existují firmy, které se objevují pouze na Internetu.

Tímto ale rozmach Internetu neskončil, neboť tento fenomén po té, co se zabydlel v kancelářích mnoha firem, obrátil svou pozornost na „obyčejné smrtelníky“ a snažil se proniknout též do jejich obývacích pokojů a pracoven. Nutno podotknout, že tato fáze trvala podstatně déle a ještě pořád neskončila, a to ze zcela zřejmých důvodů. Podle mne je ale pouze otázkou času, kdy se Internet stane jednou z nejdůležitějších (možná tou vůbec nejdůležitější) součástí našeho každodenního života.

¹ Naprostá většina dnešních uživatelů zná pouze WWW, termíny jako např. Telnet, Gopher a další málokomu něco říkají.

II. CÍLE

Cílem mé práce je vytvořit systém WWW stránek, pomocí něhož je možné publikovat vybrané informace na Internetu, popř. intranetu. Tento systém využívá Microsoft Internet Information server a technologii Microsoft Active Server Pages spolu se skriptovými jazyky JavaScript a VBScript. Jde v podstatě o vytvoření dynamických stránek, které jsou pro každého uživatele jedinečné (jejich podoba odpovídá zadaným požadavkům). Jako databázi zdrojových dat využívá tento systém data z formátu Microsoft Office, přesněji tabulkového procesoru MS Excel. Hlavním úkolem je především zajistit správnou komunikaci mezi jednotlivými částmi systému a to i v případě, že je zdrojový soubor aktualizován.

Tato práce by měla zároveň sloužit jako jakýsi průvodce pro všechny začínající uživatele, kteří mají v úmyslu vytvořit WWW stránky, které prezentují data z formátu MS Office.

III. TEORETICKÁ ČÁST

1. INTERNET

Nepochybně nejoblíbenější službou Internetu je World Wide Web, který se také stal základním pilířem obrovského nárůstu popularity Internetu v polovině devadesátých let. Nemalou měrou k tomu přispěl i první grafický prohlížeč Mosaic, který byl vytvořen v národním superpočítačovém centru (National Center for Supercomputing Applications, NCSA). Technologie WWW spatřila světlo světa již v roce 1989, a sice díky jejímu tvůrci Timothymu Berners-Leeovi, který působil v Evropském centru pro jaderný výzkum (CERN). Vývojem dnešního Webu se zabývá World Wide Web Consortium (W3C).

Technologie WWW využívá pro vlastní přenos dat protokol TCP/IP (Transmission Control Protocol/Internet Protocol), pod Web se do jednotného prostředí integrují též služby jako například FTP a Telnet. Speciálně pro WWW byly vyvinuty adresy URL (Uniform Resource Locator), protokol HTTP (HyperText Transfer Protocol) a jazyk HTML (HyperText Markup Language).

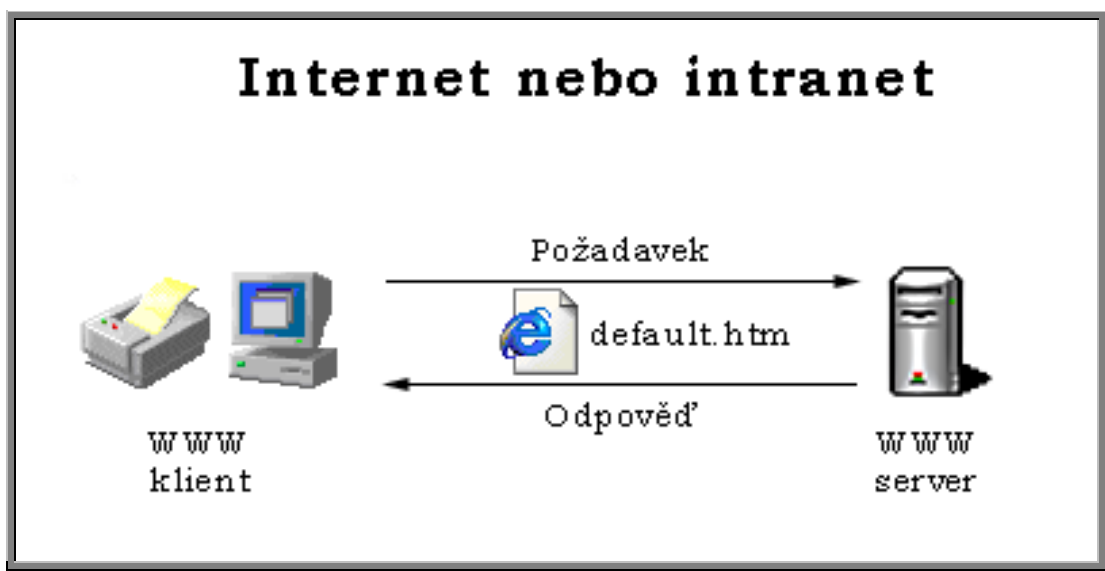
Vývojem jazyka HTML se zabývá W3C, v současnosti existuje verze 4.0. Struktura HTML stránek bývá následující:

```
<html>
<head>                                     ... hlavička
    <title>Titulek okna</title>
</head>
<body>                                     ... tělo
    <h1>Nadpis 1. úrovně</h1>
    <p>Text odstavce</p>
</body>
</html>
```

Jedná se o poměrně jednoduchý jazyk, v němž se veškeré úpravy a formátování textu provádí pomocí většinou párových příkazů, které jsou ohraničeny znaky <...> (popř. </...>). Tyto stránky mohou být buď statické nebo dynamické (ty se objevují až později). K přenosu těchto WWW stránek mezi webovým serverem

(Microsoft Internet Information Server, Apache, ...) a webovým klientem (Microsoft Internet Explorer, Netscape Navigator, ...) dochází prostřednictvím výše zmíněného protokolu HTTP – tato transakce vychází z modelu požadavek – odpověď a má obecně tento tvar: v Internetu běží velké množství počítačů – serverů, které čekají, až je někdo zavolá:

1. Prohlížeč otevře spojení s webovým serverem.
2. Prohlížeč zašle serveru požadavek.
3. Server zašle klientovi odpověď.
4. Server ukončí spojení.
5. Prohlížeč ukončí spojení.



Obrázek č. 1: Model požadavek – odpověď

1.1. STATICKÝ INTERNET

V počátcích fungování služby World Wide Web byly všechny stránky statické. To znamená, že HTML stránka již byla předem vytvořena a uložena na webovém serveru a tudíž klienti, kteří si o tuto stránku zažádali, obdrželi všichni naprosto stejnou odpověď (totožnou HTML stránku). Ovšem nelze si myslet, že statické stránky jsou jen „holé“ stránky s textem, patří sem i ty, které obsahují obrázky, zvuky apod. Komunikace probíhala přesně tak, jak ji vidíme na předchozím obrázku.

1.2. DYNAMICKÝ INTERNET

Z předchozího odstavce plyne, že webový prohlížeč byl původně pouze nástrojem, který uměl zobrazit určitý text a který dokázal přejít na definovaný hyperlink, z něhož text posléze opět načetl. Možnosti webových klientů se s postupem času různým způsobem rozšiřovaly. Lze říci, že dnešní webové prohlížeče provádí takové funkce, na které bylo v minulosti potřeba několik samostatných klientů.

Dynamický Internet se objevil s příchodem technologie CGI (Common Gateway Interface), ale mnohem důležitější roli začal plnit až s příchodem skriptových jazyků. Existují 2 základní druhy skriptování, a sice provádění skriptů na straně klienta a na straně serveru.

1.2.1. SKRIPTOVÁNÍ NA STRANĚ KLIENTA

Jak již jsem zmínil dříve, samotný jazyk HTML je interaktivní pouze v jednom případě, a to když kliknete na hypertextový odkaz (ať už v podobě textu, obrázku nebo videa), tak Vás odešle na požadovanou stránku. První společností, přesněji společnostmi, které toto změnily, byly Netscape Communications a Sun Microsystems, které vytvořily LiveScript, který umožňoval zobrazení určitých programovacích instrukcí (samozřejmě pomocí Netscape Navigatoru), které byly vloženy do HTML kódu. Později byl LiveScript přejmenován na JavaScript (byl tak trochu využit název programovacího jazyka Java).

Firma Microsoft samozřejmě nemohla zůstat pozadu, a tak vznikl skriptový jazyk Visual Basic Scripting Edition, zkráceně VBScript. Tento jazyk je užitečný hlavně pro vývojáře prostředí Windows, kteří zde mohou uplatnit své zkušenosti s Visual Basicem. A nebyla by to firma Microsoft, kdyby nevytvořila vlastní obdobu JavaScriptu nazvanou JScript.

V následujícím příkladu je ukázka, kde JavaScript a VBScript vykonávají tutéž operaci, a sice vypsání zprávy s textem „Dobrý den!“ (následující kód se zpravidla vkládá do hlavičky):

```
<script language="javascript">
  function Hlaska(){
    alert(„Dobrý den!“)
  }
```

```
</script>

<script language="vbscript">
  Sub Hlaska()
    MsgBox „Dobrý den!“
  End Sub
</script>
```

Tag `<script>` říká webovému prohlížeči, že bude následovat kód v jazyce, který je uveden u atributu *language*.

V současné době spolupracují firmy Netscape a Microsoft na novém skriptovém jazyku ECMAScript (European Computer Manufacturers Association, ECMA – Asociace evropských výrobců počítačů). ECMAScript je v podstatě JavaScript, který je přijatý jako standardní skriptový jazyk pro Web.

Skriptových jazyků existuje samozřejmě více (např. PerlScript, Python a další), ale ty se neprosadily tolik jako ty předešle jmenované.

1.2.2. SKRIPTOVÁNÍ NA STRANĚ SERVERU

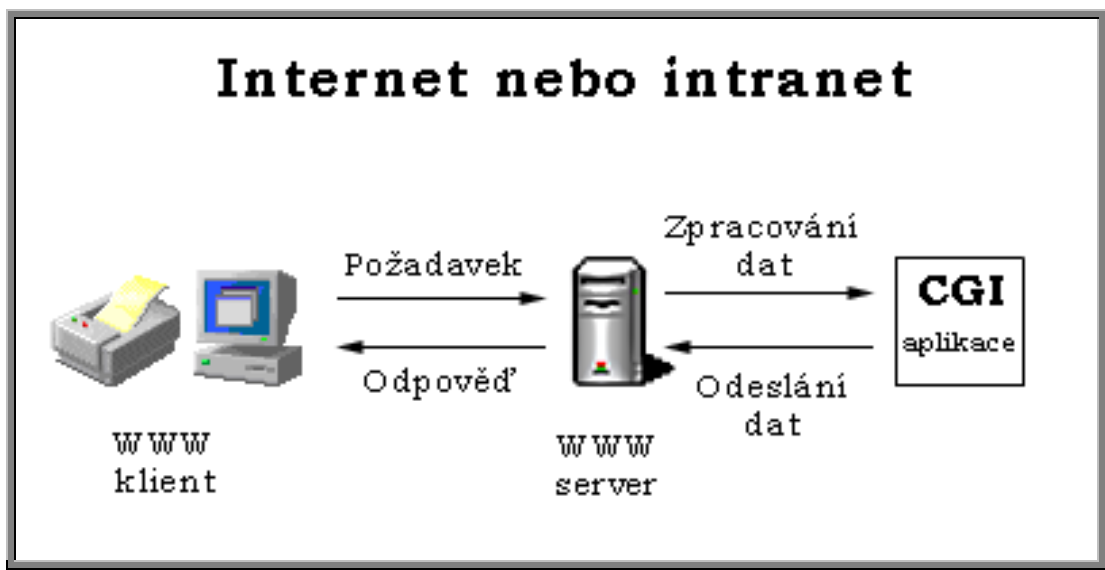
Skriptování na straně webového serveru má jednu obrovskou výhodu, a sice tu, že se uživatel nemusí starat, zda jeho WWW prohlížeče schopen zpracovat dané stránky, neboť webový server pošle prohlížeči pouze „čistý“ HTML kód.

1.2.2.1. COMMON GATEWAY INTERFACE (CGI)

Jak již jsem uvedl, jedná se o první technologii, která umožnila interaktivitu mezi webovým klientem a serverem. CGI programy jsou vykonávány WWW serverem jako odpovědi na akce uživatelů. Jejich prostřednictvím lze na straně serveru přidávat obecné funkce (např. konverze národních kódování, počítadla přístupů, vyhledávací nástroje, ...). Základní schéma činnosti CGI programu obsluhujícího formulář obsažený v HTML stránce je:

1. Jakmile uživatel aktivuje předání formuláře, webový prohlížeč posílá informace definované ve formuláři a pošle je na WWW server.
2. Server aktivuje CGI program specifikovaný parametrem *action* a předá mu data.

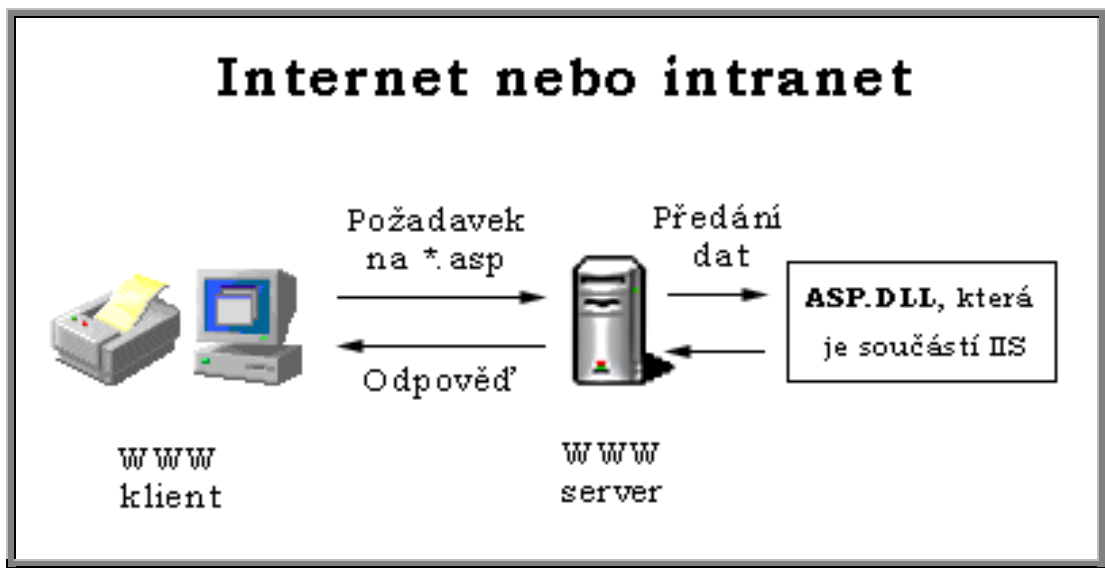
3. CGI program zpracuje data a předá výstup na standardní výstupní zařízení.
4. Výstup je zachycen WWW serverem a poslán jako výsledek klientovi, který formulář zaslal.



Obrázek č. 2: Princip fungování CGI

1.2.2.2. MICROSOFT ACTIVE SERVER PAGES (ASP)

Tato technologie je uložena v knihovně asp.dll (velikost cca. 300 kB), která je součástí Microsoft Internet Information Serveru od verze 3.0. Tato knihovna se stará o interpretaci souborů s příponou *.asp. Výsledný kód předá webovému serveru a ten jej odešle WWW klientovi.



Obrázek č. 3: Princip fungování ASP

Obrovskou výhodou technologie ASP oproti CGI je to, že ASP příkazy lze psát přímo do HTML kódu, a sice mezi značky `<%` a `%>` (popř. `<%=` a `%>`) prostřednictvím skriptového jazyka. Zajímavé je, že jako skriptový jazyk můžeme použít kterýkoliv z již dříve zmíněných jazyků – JavaScript, VBScript (ten je nastaven jako výchozí – jak jinak), Perl, JScript, C++, Javu a další. Takže, chcete-li používat místo VBScriptu např. JavaScript, musíte na první řádek ASP souboru vložit následující kód: `<%@ language="JavaScript" %>`.

Následující příklad ukazuje jednoduchý ASP soubor, který odešle webovému klientovi aktuální čas na serveru s příslušným pozdravem. Kód běžící na WWW serveru je zobrazen tučně; všimněte si též, jak je pomocí podmínky ošetřeno zobrazení minut, pokud je toto číslo menší než 10 (jinak by se totiž aktuální čas zobrazil např. 19:5 namísto 19:05):

```

<html>
<head>
  <title>ASP ukázka</title>
  <meta charset="Windows-1250">
</head>
<body>
<%

```



```

hodiny = Hour(Now)
minuty = Minute(Now)
If минуты < 10 Then
    minuty = „0“ & минуты
End If
%>
<div align="center">
Na WWW serveru je právě <%= hodiny %>:<%= минуты %> hodin.<br><br>
<%
If hodiny < 12 Then
    pozdrav = „Dobré ráno!“
Elseif hodiny < 18 Then
    pozdrav = „Dobré odpoledne!“
Else
    pozdrav = „Dobrý večer!“
End If
%>
<%= pozdrav %>
</div>
</body>
</html>

```

ASP soubor, který webový server odešle klientovi však vypadá takto:

```

<html>
<head>
    <title>ASP ukázka</title>
    <meta charset="Windows-1250">
</head>
<body>

<div align="center">
Na WWW serveru je právě 19:04 hodin.<br><br>

```

Dobrý večer!

</div>

</body>

</html>

Jak je na první pohled zřejmé, v kódu nenalezneme jiné prvky než prvky jazyka HTML, které bez problémů dokáží interpretovat všechny webové prohlížeče.

Skriptových jazyků na straně WWW serveru existuje samozřejmě mnohem více. Jedná se např. o PHP, C++, Javu, Perl a další.

2. VÝVOJ ASP A ADO

Firma Microsoft začala investovat do oblasti přístupu k datům krátce po vzniku Microsoft SQL Server verze 1.0 v roce 1989. Zájem o tuto oblast začal stoupat, ale technické prostředky byly omezené. Standard SQL byl teprve ve svých počátcích, ale již v té době bylo jasné, že to je ta „pravá revoluce“. Microsoft velmi dobře věděl, že přijetí architektury klient/server je velice výhodné a největší problém viděl v růstu možností rozhraní pro přístup k datům – ve složitosti jejich vytváření (programování), udržování, ...

Výsledkem veškerého snažení firmy bylo rozhraní Open Database Connectivity (ODBC), které, aby bylo co nejvíce efektivní, v sobě kombinuje tyto důležité rysy:

- je založeno na SQL
- podporuje širokou řadu nástrojů
- zjednodušuje implementaci pro databázové ovladače

ODBC získalo širokou podporu jako standard pro přístup k datům navzdory tomu, že mělo pár nedostatků – bylo naprogramováno jako Windows API (Application Program Interface), čímž bylo pro většinu vývojářů příliš „obtížné“. V důsledku toho vzniká nejprve technologie Data Access Objects (DAO) a pak také Remote Data Objects (RDO), které zjednodušily programovací model ODBC a zpřístupnily jej širšímu počtu programátorů.

Postupem času se objevila technologie ActiveX Data Objects (ADO), která slučuje to nejlepší z technologie DAO a RDO, z čehož plyne, že je jim velmi

podobná. Tato technologie byla poprvé představena jako součást Microsoft Internet Information Serveru (IIS) verze 3.0, přičemž je vytvořena tak, aby co nejméně zatěžovala systém. ADO (stejně jako DAO nebo RDS) využívá ODBC, pro které dnes existují řádově stovky driverů.

Technologie MS Active Server Pages se objevila také jako součást MS IIS verze 3.0 (stejně jako ADO), a jak již jsem uvedl, jedná se o skriptový jazyk na straně serveru. Proto se velice často využívá jako prostředí pro použití technologie ADO – v podstatě ve všech ASP stránkách, které přistupují k určitému typu databáze, se využívá také ADO. Mimoto ASP obsahuje několik objektů a komponent, které mají za úkol zjednodušit práci všem programátorům (blíže viz dále).

3. ALTERNATIVY K MS ASP

Jelikož technologie Active Server Pages se stává populární, začínají se objevovat verze, které lze provozovat též na jiných WWW serverech, nejenom na těch od firmy Microsoft.

Jde například o Chili!ASP od společnosti Chili!Soft, které je ze všech alternativ k ASP od Microsoftu nejvíce rozvinuté. V budoucnu by mělo podporovat vedle Windows NT například tyto operační systémy: Sun Solaris, AIX, OS/390, a to na webových serverech Apache, Netscape Enterprise, IBM ICSS a dalších.

Další společností, která se zabývá vývojem alternativní formy ASP, je Halcyon Software, která pracuje na produktu jménem Instant ASP (I-ASP). Jeho výhodou je, že je vyvíjen v jazyce Java a tudíž ho bude možno používat na většině webových serverů, které obsahují knihovny Javy. V důsledku tohoto základna použití I-ASP bude ještě širší než u produktu firmy Chili!Soft.

A jako poslední produkt bych zmínil OpenASP od společnosti ActiveScripting, jehož nejvýznamnějším rysem bude zřejmě to, že má být zdarma.

4. TECHNOLOGIE MICROSOFT ACTIVE SERVER PAGES

Jak již jsem uvedl v předchozí části, stránky ASP představují technologii, která je serverovým ekvivalentem skriptových jazyků na straně klienta. Skript běží na WWW serveru a klient přebírá veškerý jeho výstup. Skripty na stránkách ASP

obvykle ovládají určitý objekt a na výstupu generují výsledky ve formátu dynamického HTML, které jsou pro daného klienta jedinečné.

Objekty a komponenty ASP slouží ke složitějším operacím jako je například přístup k externím datům, otevírání textových souborů apod. Na tyto úkoly nám totiž samotný skriptový jazyk nestačí. Rozdíl mezi objekty a komponentami spočívá v tom, že objekty ASP jsou prvky, které jsou skriptovému jazyku vždy k dispozici (abychom je mohli použít, nemusíme je vytvářet). Naproti tomu komponenty ASP nelze použít do té doby, dokud je konkrétně nevytvoříme v kódu.

Vestavěné objekty ASP jsou tyto:

1. Objekt Application
2. Objekt Session
3. ObjektObjectContext
4. Objekt Server
5. Objekt Response
6. Objekt Request

Dříve než se začnu věnovat stručnému popisu výše zmíněných objektů ASP a poté i komponent, nemohu nezmínit jeden velice důležitý soubor, a sice GLOBAL.ASA. Tento soubor se nachází v kořenovém adresáři aplikace (pokud existuje). Je nesmírně obtížné definovat, co to vlastně internetová aplikace je. Obecně ale platí, že internetová aplikace je virtuální adresář se všemi jeho stránkami. U systému Windows aplikace začíná poklepáním na ikonu a končí kliknutím na ikonu Zavřít v pravém horním rohu. Ale jak je to vlastně na Internetu? Trvá internetová aplikace, když uživatel přeskochí z Vašich stránek na jiné a hned se zase vrátí? A co když se vrátí až za půl hodiny? A co když až za dvě? A právě proto existuje onen soubor GLOBAL.ASA, ve kterém lze určit, jaké události se provedou při začátku a konci aplikace stejně jako události při vzniku a konci práce každého uživatele (viz objekty Application a Session). K tomu slouží 4 události:

1. Application_OnStart
2. Application_OnEnd
3. Session_OnStart
4. Session_OnEnd

Z výše uvedeného plyne, že událost Application_OnStart se spouští jen jednou, zatímco událost Session_OnStart se spouští vždy, když stránku z daného adresáře žádá jiný, nový uživatel.

4.1. OBJEKT APPLICATION

Tento objekt slouží k vytváření aplikačních proměnných – to jsou proměnné, které jsou přístupné všem uživatelům aplikace. Inicializace takovýchto proměnných je vcelku jednoduchá, syntaxe je následující:

Application(“aplikační_proměnná”) = “hodnota”,

kde hodnota může být jak číslo tak textový řetězec. Objekt Application má dvě metody, a sice Lock a Unlock. Jejich existence je dána skutečností, že aplikační proměnné jsou přístupné všem uživatelům a tudíž je třeba zajistit, aby jedné proměnné nepřirazovali 2 uživatelé různou hodnotu zároveň. Ovšem jak metoda Lock tak i Unlock zamykají (odemykají) celý objekt Application a ne pouze určitou aplikační proměnnou, a proto si musíme dát pozor na zápis. Syntaxe je následující:

Application.Lock

Application(“aplikační_proměnná”) = “hodnota“

Application.Unlock

Je třeba pamatovat také na to, že při aktivaci události Application_OnEnd jsou hodnoty aplikačních proměnných vymazány.

4.2. OBJEKT SESSION

Objekt Session nám slouží k vytváření uživatelských proměnných – to jsou proměnné, které náležejí konkrétním uživatelům. Syntaxe je obdobná jako u předchozího objektu:

Session(“uživatelská_proměnná”) = “hodnota”,

kde hodnota může být opět jak číslo tak textový řetězec. Definovat uživatelské proměnné můžeme v souboru GLOBAL.ASA stejně jako v jakékoliv jednotlivé webové stránce. Odpovědí na otázku, jak si ASP pamatuje proměnné pro jednotlivé uživatele je následující – webový prohlížeč obdrží jedinečný identifikátor (Globally Unique Identifier, GUID), pomocí něhož dojde později k načtení

příslušných uživatelských proměnných. K identifikaci uživatele se použije technologie, která je obecně známa pod označením cookie.

Využití tento objekt lze například u identifikace uživatele, bezpečnostních oprávnění uživatele, individuálního nastavení uživatele (přizpůsobení vzhledu, rozlišení apod.)

Uživatelské proměnné objektu Session se zruší až tehdy, pokud uživatel restartuje svůj prohlížeč, nebo pokud je restartován webový server. Pokud uživatel přeskakuje mezi jednotlivými stránkami v aplikaci, proměnné objektu Session se nezruší, zůstanou zachovány.

4.3. OBJEKT OBJECTCONTEXT

Jde o nejmladší objekt, který se používá při využití databází. Jedná se o skripty, které se provedou jako celek, nebo se neprovedou vůbec.

Lze ho použít například při koupi knihy kreditní kartou, kdy se ověřuje, zda je kniha dostupná a zároveň zda je kontrola kreditní karty v pořádku. Pokud jedna podmínka není splněna, dojde k obnovení původního stavu.

4.4 OBJEKT SERVER

Objekt Server se používá především k vytvoření instance objektu na serveru pomocí metody CreateObject. Jakmile je instance vytvořena, lze použít vlastnosti a metody daného objektu stejným způsobem jako vlastnosti a metody objektů zabudovaných do ASP. Syntaxe je následující:

```
Set Muj_Objekt = Server.CreateObject("Nejaky_Objekt").
```

4.5. OBJEKT RESPONSE

Tento objekt zajišťuje výstupy, které webový server posílá prohlížeči. Výpis na obrazovku se provádí pomocí metody Write, ale častější je použití znaku rovnítko (=). Následující příklad provede dvěma způsoby výpis zprávy "Dobrý den!":

```
<% Response.Write "Dobrý den!" %>
```

```
<%= "Dobrý den!" %>
```

Dalším důležitým prvkem je vlastnost Expires. Pokud tuto vlastnost objektu Response nastavíme na 0 (nulu) a vložíme na první řádek (před tag <html>)

následující výraz: `<% Response.Expires = 0 %>`, prohlížeč tuto stránku nebude ukládat do paměti RAM. Díky tomu, kdykoliv tuto stránku znovu navštívíme, získáme její aktuální podobu, a nemusíme používat tlačítko Reload (Obnovit). Tato vlastnost je velmi užitečná hlavně při tvorbě ASP stránek.

4.6. OBJEKT REQUEST

Tento objekt slouží k přenosu informací od klienta na webový server. Klient odešle data pomocí formuláře, který může obsahovat textová pole, zaškrtačací políčka atd. Následující ukázka představuje velice jednoduchý internetový formulář:

```
<form action="" method="">
<input type="text" name="Jmeno">Jméno a příjmení<br>
<input type="checkbox" name="Muz">Muž<br>
<select name="Zajmy">
  <option value="Sport">Sport
  <option value="Literatura">Literatura
  <option value="Pocitace">Počítače
  <option value="Hudba">Hudba
  <option value="Divadlo">Divadlo
</select>
<input type="submit" value="Odeslat">
</form>
```

Jak je zřejmé, formulář je ohraničen klíčovými slovy `<form>` a `</form>`, přičemž nesmírně důležité jsou 2 prvky u prvně jmenovaného tagu. Prvek *action* nám říká, kde se nachází soubor, který zpracuje výstupy z formuláře (např. ASP soubor právě pomocí objektu Request) – např. `http://www.abc.cz/informace.asp`. Prvek *method* může nabývat dvou hodnot, a to *post* nebo *get*.

Metoda *get* odesílá data v adrese URL za znakem otazník (?). Z toho vyplývá, že pokud má formulář mnoho polí, adresa URL se stává nepřehledná, ale co je důležitější – počet znaků předávaných touto metodou je limitován. Pokud bychom se drželi našeho příkladu, požadovaná URL by mohla vypadat takto: `http://www.abc.cz/informace.asp?Jmeno=Jan Novák&Muz=on&Zajmy=Pocitace`.

Metoda *post* je mladší, výkonnější a více využívaná než metoda *get*. Navíc tato metoda není limitována určitým počtem znaků, neboť data se odesílají v těle požadavku a ne jako součást URL.

V samotném ASP souboru musíme nejprve načíst veškerá data odeslaná uživatelem, k čemuž použijeme právě objekt Request. Syntaxe u našeho příkladu je následující:

- u metody *get*:
jmeno = Request.QueryString("Jmeno")
muz = Request.QueryString("Muz")
If muz = "on" Then
pohlavi = "jste muž"
Else
pohlavi = "jste žena"
End If
zajmy = Request.QueryString("Zajmy")
- u metody *post*:
jmeno = Request.Form("Jmeno")
muz = Request.Form("Muz")
If muz = "on" Then
pohlavi = "jste muž"
Else
pohlavi = "jste žena"
End If
zajmy = Request.Form("Zajmy")

Existuje ještě jedna možnost, jak provést načtení dat, a to:

```
jmeno = Request("Jmeno") atd.,
```

tento způsob se ale nedoporučuje, neboť načítání je zdlouhavější (prochází se veškerá pole a při nalezení prvního odpovídajícího dojde k přiřazení hodnoty) a navíc hrozí možnost záměny.

V další části ASP souboru z našeho příkladu by mohlo být například toto:

Vaše jméno je <%= jmeno %>, <%= pohlavi %> a Vaším zájmem je (jsou)
<%= zajmy %>.

4.7. KOMPONENTY ASP

Browser Capabilities

Tato komponenta slouží ke zjištění funkcí, které podporuje webový prohlížeč uživatele.

Content Linking

Pomocí této komponenty můžeme vytvořit tabulku se seznamem všech stránek na webovém serveru.

Ad Rotator

Tato komponenta slouží k „rolování“ reklamy podle předem nastavených parametrů.

Content Rotator

Zde je situace podobná jako u předchozí komponenty, ale s tím rozdílem, že dochází k alternativnímu vkládání částí HTML.

File Access

Umožňuje přístup k místnímu a síťovému systému souborů (hlavně k textovým souborům, nejen ASP a HTML souborům).

Page Counter

Slouží k vytváření počítadla přístupů na libovolné stránce.

Permission Checker

Kontroluje povolení přístupu k určitému zdroji na místním počítači a na síti (zjistíme, zda má uživatel právo zobrazit si požadovaný soubor).

Database Access (ActiveX Data Objects, ADO)

Jedná se o nejužitečnější komponentu ASP ze všech, neboť pomocí ní můžeme číst a zapisovat do datových zdrojů ODBC. Databázi musíme nejprve zaregistrovat – v Ovládacích panelech, položce ODBC, máme dvě možnosti – Uživatelské DSN (Data Source Name) a Systémové DSN. Musíme zvolit Systémové DSN (systémový zdroj dat mohou vidět všichni uživatelé) , ne Uživatelské DSN (uživatelský zdroj dat může vidět pouze uživatel aktuálního počítače). Klikneme na tlačítko Přidat... a vybereme příslušný driver (v našem případě Microsoft Excel Driver), poté zdrojový XLS soubor včetně příslušné verze.

V ASP souboru se na námi vytvořené DSN odkazujeme například takto:

```
<%
```

```
‘ vytvoříme instanci objektu ADO
```

```
Set Nas_Objekt=Server.CreateObject(„ADODB.Connection“)
```

```
‘ otevřeme námi vytvořené DSN
```

```
Nas_Objekt.Open „DSN=Nase_DSN;UID=sa;PWD=pwd;“
```

```
... %>
```

IV. PRAKTICKÁ ČÁST

1. CO BYLO ÚKOLEM

Mým úkolem bylo vytvořit WWW stránky, které by mohly být použity v rámci podniku Karosa Vysoké Mýto a.s. (možná nejen v něm) – po zadání příslušných požadavků prostřednictvím internetového formuláře mělo dojít k zobrazení tabulek s požadovanými typy autobusů, které se prodaly v daných letech.

Zatímco v předchozí části jsem se věnoval širšímu, teoretickému popisu nejen technologie Active Server Pages od společnosti Microsoft, ale také velice stručně skriptovým jazykům JavaScript a VBScript, v této části bych se chtěl věnovat mnohem konkrétněji samotným WWW stránkám a všemu, co s jejich vznikem souvisí. Jinými slovy bych tuto část mé bakalářské práce prezentoval jako návod či určitého průvodce, v jakých krocích jsem postupoval, neboť se mi nikde v literatuře ani na Internetu nepodařilo nalézt takový materiál, který by se přesně tímto zabýval. Nyní mám na mysli především to, jak pomocí ASP „vstoupit“ do databáze Microsoft Excel, neboť ve všech materiálech, které jsem našel a z kterých jsem čerpal, se využívá databáze programu Microsoft Access.

2. POTŘEBNÉ SOUČÁSTI

K tomu, aby mohly být vytvořeny a provozovány ASP stránky, je nezbytný Microsoft Internet Information Server nebo jeho obdoba Personal Web Server. Toto samozřejmě neplatí na 100 %, blíže viz část pojednávající o alternativách k MS ASP.

K samotnému vytvoření ASP stránky stačí obyčejný Poznámkový blok, který je standardní součástí instalace každých Windows. Já bych ale doporučoval nějaký specializovaný editor jazyka HTML, ale osobně uznávám jen ty editory, ve kterých se zdrojový kód musí psát „ručně“. V této kategorii je zřejmě nejlepší HomeSite (nyní ve verzi 4.5) od firmy Allaire, kterému ale velice slušně sekunduje 1st Page 2000 od firmy Evrsoft, který má oproti předchozímu programu jednu obrovskou výhodu, a sice tu, že je zdarma (kategorie freeware). Existuje ještě druhá skupina webových editorů, které nepředpokládají znalost jazyka HTML (natož nějakého skriptového jazyka) – zde bych jako vůdce této kategorie jmenoval FrontPage od firmy Microsoft (nyní ve verzi 2000). Ale záleží pouze na

vás, kterou kategorii si zvolíte – každá má své výhody a nevýhody, které jsem shrnul do následující tabulky.

„Inteligentní“ webové editory (např. MS FrontPage 2000)	
Klady	Zápory
– není nutná znalost HTML	– soubory jsou mnohem větší
– stačí pouze „klikat“	– zdrojový kód je nepřehledný
– tvorba stránek je velice rychlá	– možnosti tvorby stránek jsou omezené

Tabulka č. 1: Výhody a nevýhody „inteligentních“ editorů

Výhody a nevýhody druhé skupiny editorů logicky vyplývají z výše uvedené tabulky.

Další nezbytnou součástí, pokud se má přistupovat k XLS tabulkám, je samozřejmě program MS Excel (nyní ve verzi 2000 – jako součást MS Office 2000), nebo alespoň tabulky, které byly v tomto programu vytvořeny (nebo v jeho formátu uloženy). Toto samotné ale nestačí, neboť musí být přesně stanoveno, ke které tabulce se má v daném okamžiku přistupovat. U databáze MS Access je toto velice jednoduché – samotný MDB soubor obsahuje tabulky (také dotazy, sestavy atd.), které mají určitý název a na ty se pak stačí odvolávat v ASP stránce. Toto ale u XLS tabulek samo o sobě nelze, a proto je nutné takovýto název vytvořit. Tato operace se provede následovně:

- označí se tabulka, ke které se má přistupovat
- v položce Vložit, Název, Definovat ... se napíše název, který nesmí začínat číslicí

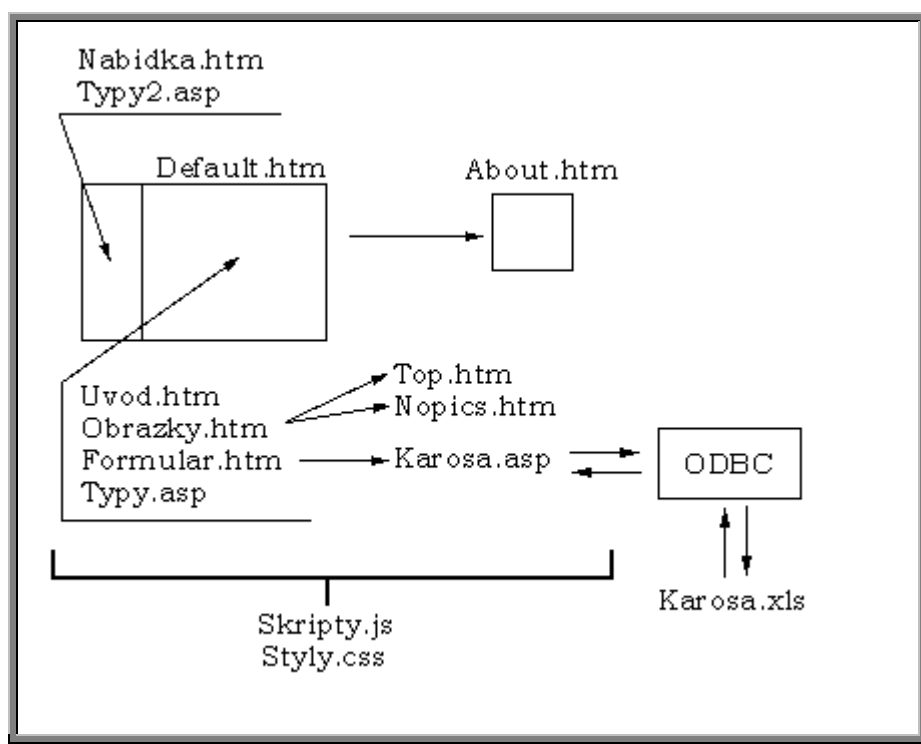
Pokud je třeba přistupovat k více tabulkám (jako v mém případě), nejenom k jedné, celý postup se musí opakovat pro každou tabulku zvlášť a na závěr je nezbytné XLS tabulku uložit.

Dalším krokem je identifikace dané databáze v aplikaci ODBC (v Ovládacích panelech), kterou jsem již podrobně popsal v části **Komponenty ASP**, a proto se k ní nehodlám vracet.

Poslední, ale neméně důležitou součástí jako ty předešlé, je webový prohlížeč. V současnosti existuje několik WWW klientů – rozšířené jsou ale pouze tyto dva: Microsoft Internet Explorer (nyní ve verzi 5.5 – je standardní součástí instalace každých Windows) a Netscape Navigator (nyní ve verzi 4.7). Existují

ještě některé další prohlížeče (např. Opera, Hot Java Browser a další), ale ty se vyskytují spíše vzácně. Ale zpět k prvním dvěma jmenovaným – člověku „neznalému“ by se mohlo zdát, že je zcela jedno, zda používá prohlížeč od společnosti Microsoft nebo Netscape, ale jak pozná v části, ve které se věnuji popisu tvorby HTML formuláře, je velice těžké vytvořit WWW stránku, která by vypadala a chovala se naprosto stejně jak v MS IE tak NN (korektně se stránka zobrazí – a to ne vždy, pokud používáte samotné HTML, ale pokud použijete skriptový jazyk nebo objekty DHTML (Dynamického HTML), výsledek je v každém prohlížeči odlišný).

3. SCHÉMA WWW STRÁNEK



Obrázek č. 4: Schéma WWW stránek²

4. TVORBA WWW STRÁNEK

V této části bych se chtěl věnovat samotným WWW stránkám, ale samozřejmě ne všem, ale pouze těm, které jsou čímsi zvláštní, odlišné anebo problematické. Nemíním zde ztrácet čas popisem souborů jako Nabidka.htm,

² Se souborem Karosa.xls jsou pomocí ODBC propojeny všechny ASP soubory, nejenom Karosa.asp

Default.htm a dalších, neboť se jedná o standardní HTML stránky, jak je možné zjistit v Příloze.

V podstatě se chci věnovat následujícím souborům:

- Karosa.xls
- Styly.css
- Skripty.js
- Formular.htm
- ASP soubory

4.1. KAROSA.XLS

Jedná se o soubor MS Excel 97, který má deset listů (pojmenované 1990 až 1999). Na každém listu se nachází tabulka, ve které jsou zachyceny prodeje jednotlivých modelů autobusů za příslušný rok. Je důležité, aby první řádky tabulek byly nějakým způsobem odlišeny (tučné písmo, větší velikost písma apod.), aby bylo zřejmé, že se jedná o hlavičky sloupců.

Typ	ČR	SR	Ostatní	Celkem
B 731	127	18	100	245
B 732	336	179	81	596
C 734	811	451	153	1415
C 735	388	69	99	556
C 744	110	39	1	150
LC 736	182	44	25	251
Celkem	1954	800	459	3213

Tabulka č. 2: Karosa.xls – ukázka

4.2. STYLY.CSS

K tomuto souboru pouze pár slov – jedná se o velmi užitečný soubor, ve kterém lze definovat veškeré formátování textu, pozadí, nadpisů atd., ale i vytvářet nové prvky (např. stínované písmo pomocí určení absolutních pozic textu či pomocí vrstev). Nechci zde podrobně popisovat, jak se provádí zápis, neboť to lze nalézt v každé referenci CSS (Cascading Style Sheets) nebo HTML 4.0. Jedině, co bych zde chtěl zmínit k používání CSS je to, že se na ně odvolává v HTML souborech pomocí klíčových slov *class* nebo *id*. V následujícím příkladě je vidět, jak se používají.

```
<html>
<head>
<title>CSS ukázka</title>
```

```

<meta charset="Windows-1250">
<style type="text/css">
    .modre {color: blue;}
    #cervene {color: red;}
</style>
</head>
<body>
<p class="modre">Toto je napsáno modře.
<p id="cervene">Toto je napsáno červeně.
</body>
</html>

```

Lze také vytvořit soubory, které obsahují pouze CSS prvky (to jsou ty mezi tagy `<style>` a `</style>`). Základní podmínkou použití těchto souborů v HTML nebo ASP stránkách je to, že musí být identifikovány (načteny) v hlavičce, a tuto operaci lze provést dvěma způsoby:

- `<link rel="stylesheet" type="text/css" href="Styly.css">`
- `<style type="text/css">`

```

    @import url("Styly.css")
</style>

```

Zdrojový soubor (Styly.css) musí být zadán spolu s absolutní nebo relativní cestou. Pokud porovnáme oba dva způsoby (implementaci a načtení), je zřejmé, že druhý způsob je daleko výhodnější. Nevýhodou prvního způsobu je to, že se na takto definované styly lze odvolávat pouze v tom souboru, kde jsou definovány. Pokud jsou ale uvedeny v samostatném souboru, lze se na ně (po jejich identifikaci) odkazovat v jakémkoliv souboru. Jinými slovy, styly stačí napsat pouze jednou do CSS souboru a ne do všech HTML (ASP) souborů.

4.3. SKRIPTY.JS

O tomto souboru lze říci, že je to obdoba předchozího souboru, ale s tím rozdílem, že namísto stylů obsahuje funkce ve skriptovém jazyce JavaScript (vyznám zásadu, že na straně klienta je využíván skriptový jazyk JavaScript a na straně serveru VBScript). Jinými slovy, veškeré funkce, které jsou zde

vytvořeny, lze použít v jakémkoliv HTML nebo ASP stránce, ve které je provedena identifikace souboru Skripty.js, a to následujícím způsobem:

```
<script language="javascript" type="text/javascript" src="Skripty.js">
</script>
```

Zdrojový soubor (Skripty.js) musíme opět uvést s absolutní nebo relativní cestou.

4.4. FORMULAR.HTM

Kupodivu největší problémy jsem neměl s vytvořením ASP souborů, ale právě s vytvořením formuláře. Potíže nesouvisely s jednotlivými poli formuláře, ale s jednotlivými prohlížeči. V této stránce jsem totiž vytvořil zaškrtačací políčko *Jensuma*, které mělo za cíl zobrazit ve volaném ASP souboru pouze hlavičky tabulek a jeden jediný řádek, a to celkové sumy prodaných autobusů v jednotlivých teritoriích. Z toho důvodu jsem musel zajistit, že pokud je toto políčko zaškrtnuto, budou ignorovány veškeré změny v políčkách *Typ*, *Teritorium*, *Sumatyp* a *Sumater*.

The image shows a web form with the following elements:

- Roky:** A list box containing the years 1990, 1991, and 1992.
- Dle:** A label for the sorting criteria.
- Tabulky řadit dle:** A dropdown menu currently set to "Typu".
- Razeni:** Radio buttons for "Vzestupně" (selected) and "Sestupně".
- Jensuma:** A checkbox labeled "Pouze suma (za typy i teritoria)".
- Typ:** An empty text input field.
- Sumatyp:** A checked checkbox labeled "Suma za jednotlivé typy (ze rok)".
- Teritorium:** A list box containing "Česká republika", "Slovenská republika", and "Ostatní".
- Sumater:** A checked checkbox labeled "Suma za jednotlivá teritoria (za rok)".
- Buttons:** "Odeslat formulář" and "Všechno vymazat".

Obrázek č. 5: Internetový formulář

Nyní mohu říci, že až ve třetí verzi tohoto formuláře se mi tuto situaci podařilo uspokojivě vyřešit. Problémy s políčkem *Jensuma* vůbec nesouvisely s jazykem HTML nebo JavaScript, ale jak již jsem zmínil výše – s prohlížeči. Přesněji řečeno s jedním konkrétním prohlížečem, a to Netscape Navigátorem (všechny tři verze mého formuláře fungovaly naprosto bezchybně pouze v MS Internet Exploreru). Dokonce jsem byl již tak „zoufalý“, že jsem políčko *Jensuma* chtěl úplně zrušit.

4.4.1. FORMULAR.HTM – 1. VERZE

Jak již jsem naznačil, v počátcích mé tvorby těchto WWW

Pouze suma (za typy i teritoria): Ano Ne

stránek jsem měl trochu obavy, jak zvládnou vytvoření vlastních ASP stránek. Tyto obavy se však ukázaly jako zcela zcestné, neboť největší problémy jsem měl právě s vytvořením formuláře, přesněji s jeho korektním fungováním v prohlížečích, tedy v jednom konkrétním, a to v NN (jak již jsem uvedl, MS IE si bez problémů poradil se všemi třemi verzemi mého formuláře).

Můj původní záměr (uskutečněný v této verzi) byl jednoduchý a pro mne zcela jasný – jednoduše využiji vlastností a událostí jednotlivých objektů DHTML (blíže viz Příloha A). To znamená, že pokud kliknete na políčko *Jensuma_ano*, provede se skript, který je k tomuto poli přiřazen pomocí události *onclick*. Skript vypadá následovně:

```
function skryti() {  
    document.forms[0].Jensuma[1].checked = false;  
    document.forms[0].Typ.disabled = true;  
    document.forms[0].Sumater.disabled = true;  
    document.forms[0].Sumatyp.disabled = true;  
    document.forms[0].Teritorium.disabled = true;  
}
```

Je zřejmé, že zde jde o jednoduchou podmínku (obdobná podmínka musí samozřejmě existovat též pro pole *Jensuma_ne* – funkce *odkryti()* je stejná, ale místo hodnot *true* nabývá *false* a naopak), která po kliknutí na pole *Jensuma_ano*

přiřadí určitým polím vlastnost *disabled*. Zatímco MS IE tuto vlastnost podporuje, tak NN ji s úspěchem ignoroval.

V souvislosti s tímto musím upozornit ještě na jeden problém, a sice také velice důležitý – to se týká všech verzí formuláře, nejenom této jedné. Jde zde o to, že pokud vše funguje tak, jak má (pokud kliknete na určité políčko a ostatní se zneprístupní), tak to ještě neznamena, že je vše v pořádku. V momentě, kdy je formulář odeslán a zobrazí se požadovaná stránka, je ještě vše v pořádku, ale pokud se klikne na tlačítko *Zpět*, může nastat nepříjemná situace – a to, že políčko *Jensuma_ano* je zaškrtnuto, ale ostatní políčka jsou dostupná. Zde není chyba v podmínkách, ale tato situace je způsobená tím, že v momentě, když se klikne na tlačítko *Zpět*, pak některým prvkům zůstanou vlastnosti, které měli v momentě odeslání, avšak u některých dojde k opětovnému načtení jejich výchozího stavu. Jinými slovy – pokud se do pole *Typ* napíše např. „A“, pak po navrácení se zpět bude hodnota tohoto pole opět „A“. Ale pokud se přiřadí pomocí již výše uvedeného skriptu nějakému objektu vlastnost *disabled*, pak po kliknutí na tlačítko *Zpět* dojde k načtení původních vlastností (tím pádem daný objekt vlastnost *disabled* nemá, ale políčko, které mu tuto vlastnost přiřazuje, je zaškrtnuté). Tento problém jsem vyřešil, a sice takto – do tagu `<body>` lze přidat událost *onload*, která volá specifikovanou funkci vždy, když dojde k načtení stránky. V mém případě (u této verze formuláře) se volala funkce *odkryti()*. Jedná se o stejnou funkci, která je volána, pokud uživatel klikne na políčko *Jensuma_ne*.

4.4.2. FORMULAR.HTM – 2. VERZE

Jelikož 1. verze tohoto souboru nefungovala korektně v prohlížeči NN, přesto že jsem namísto vlastnosti *disabled* použil *readonly* (opět MS IE podporuje obě tyto vlastnosti), rozhodl jsem se vyřešit tuto situaci pomocí stylů (CSS). Každému prvku (objektu DHTML) lze totiž přiřadit vlastnost *style* – díky ní se nám otevře cesta ke všem prvkům CSS. Následující příklad ukazuje, jak se změní barva nadpisu, pokud na něj najedete myší (k elementu se přistupuje pomocí jeho *id*):

```
<html>
<head>
  <title>CSS – ukázka</title>
  <meta charset="Windows-1250">
```

```

<script language="javascript">
function css(){
    document.all.nadpis.style.color = „red“;
}
</script>
</head>
<body>
<h2 id="nadpis" onmouseover="css()">Zadejte požadavky:</h2>
</body>
</html>

```

Ve formuláři jsem použil prvek *display* s hodnotami *none* (skrytí) a *inline* (zobrazení). Výsledek byl uspokojující, i když ne tak elegantní jako u verze číslo 1. Ostatně to lze posoudit z následujícího obrázku.

The image shows a web form with a white background and a black border. It contains two rows of text. The first row has the label 'Typ:' followed by the text 'Suma za jednotlivé typy (ze rok)'. The second row has the label 'Teritorium:' followed by the text 'Suma za jednotlivá teritoria (za rok)'. At the bottom of the form, there are two buttons: 'Odeslat formulář' and 'Všechno vymazat'.

Obrázek č. 6: Formulář se skrytými poli

Co říci závěrem k této verzi? Snad jen to, že i zde bylo nutné vytvořit funkci, která byla volána událostí *onload* a také to, že tento způsob podporoval opět pouze MS IE. A tak jsem musel vytvořit 3. verzi formuláře, která fungovala bezchybně i v NN.

4.4.3. FORMULAR.HTM – 3. VERZE

V této verzi jsem se vrátil zpět k vlastnostem jednotlivých prvků a omezil jsem se opravdu pouze na ty, které podporují oba prohlížeče (hlavně ten jeden).

Nakonec se mi podařilo vytvořit takovou podmínku, která fungovala správně i v NN a tak jsem ji s mírnými úpravami úspěšně použil u všech prvků formuláře pomocí události *onclick*. V podstatě jde o to, že pokud uživatel klikne například na pole *Sumatyp*, tak skript zjistí, zda je zaškrtnuté pole *Jensuma*. Pokud ne, políčko *Sumatyp* se zaškrtně/odškrtně, pokud ale ano, pak se zobrazí varovná hláška se slovy: „Nejprve musíte odznačit položku ‚Pouze suma!‘“. Navíc jsem tuto verzi formuláře vylepšil (podle mne ne nevýznamně), a to takto: pokud např. do políčka *Typ* uživatel napíše např. „B 731“ nebo cokoliv jiného a poté klikne na pole *Jensuma*, tak se hodnota pole *Typ* změní na „Žádný“ (a nelze ji změnit). Pokud uživatel znovu klikne na pole *Jensuma* (odznačí jej), políčko *Typ* nezůstane prázdné, ale objeví se zde hodnota, která tu byla před zaškrtnutím pole *Jensuma* (obdobně se chovají i ostatní pole). Stejným způsobem toto funguje, pokud uživatel formulář odešle a vrátí se na něj zpět. Jinými slovy, musel jsem ještě upravit funkci, která se volá při načítání stránky pomocí události *onload*. Jak jsem tuto situaci vyřešil? To můžete zjistit nahlédnutím do zdrojového kódu v Příloze.

4.5. KAROSA.ASP

Jedná se o hlavní ASP soubor – ten, který zpracovává data z formuláře, a proto se mu hodlám věnovat podrobně. V následujícím textu vždy uvedu část zdrojového kódu ASP souboru spolu s popisem (kurzívou za apostrofem). Samozřejmě vynechám věci, které se opakují (podmínky apod.) pro různá pole formuláře.

```
<%  
jensuma = Request.Form(„Jensuma“)  
roky = Request.Form(„Roky“)  
typ = Trim(Request.Form(„Typ“)) atd.
```

‘ načtou se hodnoty pomocí objektu Request; funkce Trim odstraňuje počáteční a koncové mezery; jelikož roků lze najednou vybrat více, musí být následně rozděleny

```
rok = Split(roky, „, „, -1, 1)
```

‘ následuje velké množství podmínek, uvedu jednu za všechny

```
If jensuma = „on“ Then
```

```
    typ = „_x_“
```

```

        vypistyp = „Žádný“
        sumatyp = „on“
        sumater = „on“
        tera = „ČR, SR, Ostatní“
    End If
    ' vytvoří se instance objektu ADO
    Set Propojeni=Server.CreateObject(„ADODB.Connection“)
    ' otevře se propojení pomocí námi definovaného DSN (UID představuje
    uživatele a PWD heslo)
    Propojeni.Open „DSN=Karosa;UID=sa;PWD=pwd;“
    ' jelikož roků může být více, musí být zaveden cyklus (LBound představuje
    nejnižší index (vždy nula) a UBound nejvyšší – POZOR: pro např. 5 prvků
    nabývají indexy hodnot 0 až 4!)
    For i = LBound(rok) To UBound(rok)
        ' dále je vytvořen SQL dotaz
        dotaz=“SELECT * FROM „&rok(i)&“ WHERE („&rok(i)&“.Typ LIKE
        „&typ&“%) AND („&rok(i)&“.Typ NOT LIKE ‚Celkem‘) ORDER BY
        „&rok(i)&“.„&dle&“ „&razeni&“;“
        ' a vytvoří se ještě jeden SQL dotaz – ten se vypíše, pokud je zaškrtnuto
        pole Sumater
        dotaz_suma=“SELECT * FROM „&rok(i)&“ WHERE („&rok(i)&“.Typ LIKE
        ‚Celkem‘) ORDER BY „&rok(i)&“.Typ ;“
        ' dále se vytvoří 2 objekty RecordSet objektu Connection – ty představují
        záznamy vrácené dotazem
        Set vysledek=Propojeni.Execute(dotaz)
        Set vysledek_suma=Propojeni.Execute(dotaz_suma)
        ' dále je třeba zajistit, aby došlo k vypsání zprávy, pokud je objekt vysledek
        prázdný (to říká vlastnost EOF) – např. nebyl nalezen odpovídající typ
        If vysledek.EOF AND jensuma<>“on“ Then %>
            Požadavek na <%= Left(rok(i),3) %> <b><%= Right(rok(i),4) %></b>: Nebyl
            nalezen odpovídající záznam.<br><br>

```

' pokud tato podmínka není splněna, provede se vypsání příslušných záznamů ve formě tabulky

```
<% Else %>
```

```
<table border="3" cellpadding="3" cellspacing="3">
```

```
<% Do Until vysledek.EOF %>
```

```
<tr>
```

```
<td><i>&nbsp;<%= vysledek(„Typ“) %></i></td> atd.
```

' přesun na další záznam

```
<% vysledek.MoveNext %>
```

' ukončí se cyklus Do

```
<% Loop %>
```

' vypíše se řádek s celkovými součty za jednotlivá teritoria (pokud si jej uživatel vyžádal)

```
<% If Sumater = „on“ Then %>
```

```
<tr>
```

```
<td><b>&nbsp;<%= vysledek_suma(„Typ“) %></b></td> atd.
```

```
</tr>
```

```
<% End If %>
```

' ukončí se cyklus pro daný rok a nastane přesun na další

```
<% Next %>
```

' a nakonec je třeba objekt propojení uzavřít

```
<% Propojeni.close %>
```

Výsledek pak může vypadat např. takto:

Výsledky:			Vaše požadavky:
			Typ: Všechny
			Rok: 90
			Řazení: Typ, Vzestupně
			Teritorium: ČR
			Celkem typy: Ano
			Celkem teritoria: Ano
Prodej autobusů Karosa za rok 1990			
Typ	ČR	Celkem	
B 731	127	245	
B 732	336	596	
C 734	811	1415	
C 735	388	556	
C 744	110	150	
LC 736	182	251	
Celkem	1954	3213	

Obrázek č. 7: Zobrazení výsledků

4.6. TYPY.ASP

Tento soubor slouží k vypsání typů autobusů, které se za dané roky prodaly. Lze říci, že se jedná o zjednodušenou obdobu předchozího souboru. Za zmínku stojí pouze 2 věci:

- tento soubor obsahuje jak formulář, tak i zobrazení výsledků – z toho plyne, že volá sám sebe
- typy autobusů se zobrazují v řádcích tabulky, která má vždy 5 sloupců; jelikož ale počet typů není vždy beze zbytku dělitelný 5, bylo nutné tuto situaci ošetřit jednoduchou podmínkou

4.7. TYPY2.ASP

Tento soubor je jakousi „třešničkou na dortu“ – objeví se, kdykoliv dojde ke změně ve výběru roků ve formuláři, a sice místo levého rámu (soubor

Nabidka.htm). Takže pokud uživatel klikne např. na rok 1990, pak se v levém rámu objeví tabulka s jednotlivými typy, které se v roce 1990 prodávali. Pak stačí pouze kliknout na požadovaný typ – ten funguje jako odkaz, který vloží do pole *Typ* onen model autobusu. Toto samozřejmě funguje i pro více roků (tzn. můžete vybrat více než jeden rok a poté se zobrazí všechny roky s příslušnými typy autobusů). Samozřejmě pokud je zaškrtnuto políčko *Jensuma*, zobrazí se již zmiňované výstražné okno se zprávou.

To, co jsem výše popsal, zní jednoduše, ale bylo velice těžké přijít na to, jak tento nápad zrealizovat. Musel jsem nějakým způsobem zajistit, aby když uživatel klikne na nějaký rok, došlo jakýmsi způsobem k odeslání tohoto roku dané stránce *Typy2.asp* a k následnému zobrazení dané tabulky. U celého formuláře je to jednoduché – stačí kliknout na tlačítko odeslat. Ale to tady nepřicházelo v úvahu (ne že by to nešlo takto provést, ale toto řešení by bylo málo „elegantní“). A proto jsem se rozhodl vytvořit takový skript (který je volán událostí *onchange* u prvku *Roky*), který by zjistil, které roky jsou vybrány a poté by je odeslal v příslušné formě stránce *Typy2.asp*. To znamená, že jsem musel simulovat metodu *get*, neboť ta posílá data v URL za znakem *?* (naproti tomu metoda *post* data posílá v těle požadavku). Onen skript, který se stará o odeslání správné URL vypadá takto:

```
function fce(obj) {
    var roky = „“;
    for (i = 0; i < obj.length; i++) {
        if ((obj[i].selected) ||
            (obj[i].checked)) {
            roky = roky + obj[i].value + „“;
        }
    }
    n = roky.length;
    n = n - 1;
    roky = roky.substr(0,n);
}
```



```
url = „Typy2.asp?Roky=“ + roky;  
parent.Nabidka.location.href = url;  
}
```

Ještě dvě poznámky na závěr:

1. Nejprve jsem jako oddělovač jednotlivých roků použil čárku s mezerou (,) tak, jak to dělá metoda *post*, ale s tímto si opět neporadil NN. Po nějaké době (když jsem již vyzkoušel snad vše) jsem zkusil použít pouze čárku bez mezery a kupodivu, vše fungovalo bez chyby nejen v MS IE, ale i NN.
2. Velkým problémem se vcelku nečekaně stal onen odkaz, který požadovaný model autobusu vyplnil do pole *Typ*. On je to totiž odkaz nikam, a jak takový odkaz vytvořit, to se člověk nikde nedočte. Pokud se použije atribut *nohref*, daný text se nechová jako hypertextový odkaz. Pokud se použije odkaz na návěstí (Odkaz), NN se po kliknutí na takovýto odkaz vrátí na začátek stránky, ale MS IE zůstane tam, kde je (což bylo mým cílem). Nakonec jsem tuto situaci vyřešil velmi podivně, ale hlavní je, že to takto funguje – .

V. ZÁVĚR

Internet umožnil otevřít obrovský prostor pro globální komunikaci lidí všech možných zájmů, s neuvěřitelnou lehkostí stírá veškeré geografické vzdálenosti a je nepochybně jediným prostředkem, který toto dokáže.

Jak již jsem několikrát zmínil, svět Internetu se nám všem mění před očima – co bylo včera moderní a pokrokové, se dnes jeví jako archaické a dávno překonané. Stejně jako samotný Internet se mění i technologie, které jej vytvářejí – základem je sice pořád HTML, ale objevuje se veliké množství jeho obdob jako například DHTML, CSS, XML apod., které původní jednoduchý hypertext ani zdaleka nepřipomínají. Lze říci, že každý webmaster se musí neustále učit a zdokonalovat, aby udržel krok s konkurencí. Mohu bez nadsázky prohlásit, že dnešní web se programuje – objevuje se obrovské množství programovacích jazyků, které se stále více na Internetu prosazují. Například JavaScript, jehož znalost je dnes již považována za samozřejmost, dále VBScript, Active Server Pages, jeho obdoba pro unixové servery PHP, ... Navíc většina opravdu velkých webů je dnes již téměř vždy generována z databází.

VI. SEZNAM POUŽITÉ LITERATURY

1. Corning, Michael aj. Working with Active Server Pages. Que Corporation, 1997.
2. Derfler, Frank J. a Freed, Les. Jak pracují sítě. Brno: Unis, 1994.
3. Hillier, Scot a Mezick, Daniel. Programování Active Server Pages. Brno: Computer Press, 1998.
4. Hlavenka, Jiří aj. Vytváříme WWW stránky a spravujeme moderní web site. Praha: Computer Press, 1997.
5. Horný, Stanislav. Návrh a tvorba prezentací WWW. Praha: VŠE, Fakulta informatiky a statistiky, Katedra systémové analýzy, 1998.
6. Mikle, Pavel. DHTML – dynamické HTML, referenční příručka. Brno: Unis, 1997.
7. Naik, Dilip C. Internet – standardy a protokoly. Brno: Computer Press, 1999.
8. Renda, Miroslav. Český Internet a Internet Explorer 4 – podrobný průvodce začínajícího uživatele. Praha: Grada Publishing, 1998.
9. Šmrha, Pavel a Rudolf, Vladimír. Internetworking pomocí TCP/IP. České Budějovice: KOPP, 1994.
10. Weissinger, A. Keyton. ASP in a Nutshell: A Desktop Quick Reference. O'Reilly & Associates, Inc. 1999.

VII. PŘÍLOHY

Příloha A: Dynamické HTML

Příloha B/1: Zdrojový kód souboru Default.htm

Příloha B/2: Zdrojový kód souboru Nabidka.htm

Příloha B/3: Zdrojový kód souboru Formular3.htm

Příloha B/4: Zdrojový kód souboru Karosa.asp

Příloha B/5: Zdrojový kód souboru Typy.asp

Příloha B/6: Zdrojový kód souboru Typy2.asp

Příloha B/7: Zdrojový kód souboru Styly.css

Příloha B/8: Zdrojový kód souboru Skripty.js

Příloha C: Údaje pro knihovnickou databázi

Příloha A

Dynamické HTML

Množina objektů:

- window
 - location
 - frames
 - history
 - navigator
 - event
 - visual
 - document
 - ◆ links
 - ◆ anchors
 - ◆ images
 - ◆ forms
 - ◆ applets
 - ◆ embeds
 - ◆ plugins
 - ◆ frames
 - ◆ scripts
 - ◆ all
 - ◆ selection
 - ◆ body

Standardní vnitřní události:

- onload
- onunload
- onclick
- ondblclick
- onmousedown
- onmouseup
- onmouseover
- onmousemove
- onmouseout
- onfocus
- onblur
- onkeypress
- onkeydown
- onkeyup
- onsubmit
- onreset
- onselect
- onchange

Příloha B/1

Zdrojový kód souboru Default.htm

```
<html>
<head>
  <title>Autobusy Karosa a.s. Vysoké Mýto</title>
  <meta charset="Windows-1250">
  <link rel="stylesheet" type="text/css" href="Style.css">
  <script language="javascript" type="text/javascript" src="Skripty.js">
  </script>
</head>

<frameset cols="108,*" frameborder="no" framespacing="0">
  <frame src="Nabidka.htm" name="Nabidka" frameborder="0"
framespacing="0" marginheight="0" marginwidth="0" scrolling="auto" noresize>
  <frame src="Uvod.htm" name="Obsah" frameborder="0" framespacing="0"
marginheight="0" marginwidth="0" scrolling="auto" noresize>
</frameset>

<noframes>
<body>
<div align="center">
Je mi líto, ale tyto stránky používají rámce (frames).
<br>
Budete si asi muset pořídit prohlížeč, který je podporuje.
</div>
</body>
</noframes>

</html>
```

Příloha B/2

Zdrojový kód souboru Nabidka.htm

```
<html>
<head>
  <title>Nabídka</title>
  <meta charset="Windows-1250">
  <link rel="stylesheet" type="text/css" href="Styly.css">
  <script language="javascript" type="text/javascript" src="Skripty.js">
  </script>
</head>

<body background="Obrazky/Nabidka.gif">
<div align="center">

<table width="90%" border="0">
<tr>
  <td>
    <br>
    
    <br><br><br>
    <a href="Uvod.htm" target="Obsah" onmouseover="statbar('Úvodní
stránka');return true" onmouseout="statbar(' ');return true" title="Úvodní
stránka">Úvodní<br>stránka</a><br><br>
    <a href="Obrazky.htm" target="Obsah" onmouseover="statbar('Obrázky
autobusů Karosa');return true" onmouseout="statbar(' ');return true" title="Obrázky
autobusů Karosa">Obrázky<br>autobusů</a><br><br>
    <a href="Formular.htm" target="Obsah" onmouseover="statbar('Formulář
pro Váš výběr');return true" onmouseout="statbar(' ');return true" title="Formulář
pro Váš výběr">Formulář1</a><br><br>
```

```
<a href="Formular2.htm" target="Obsah" onmouseover="statbar('Formulář
pro Váš výběr');return true" onmouseout="statbar(' ');return true" title="Formulář
pro Váš výběr">Formulář2</a><br><br>
```

```
<a href="Formular3.htm" target="Obsah" onmouseover="statbar('Formulář
pro Váš výběr');return true" onmouseout="statbar(' ');return true" title="Formulář
pro Váš výběr">Formulář3</a><br><br>
```

```
<a href="Typy.asp" target="Obsah" onmouseover="statbar('Přehled
dostupných typů autobusů');return true" onmouseout="statbar(' ');return true"
title="Přehled dostupných typů autobusů">Dostupné<br>autobusy</a><br><br>
```

```
<a href="javascript: otevrit('About.htm')" onmouseover="statbar('Pár slov o
mně ...');return true" onmouseout="statbar(' ');return true" title="Pár slov o mně
...">About ...</a><br><br>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
<map name="Logo_male">
```

```
<area shape="poly" alt="WWW stránky podniku KAROSA a.s."
coords="24,1,23,33,55,32,41,18,56,2" href="http://www.karosa.cz" title="WWW
stránky podniku KAROSA a.s." target="Obsah" onmouseover="statbar('WWW
stránky podniku KAROSA a.s. ');return true" onmouseout="statbar(' ');return true">
```

```
<area shape="rect" alt="WWW stránky podniku KAROSA a.s."
coords="0,40,78,54" href="http://www.karosa.cz" title="WWW stránky podniku
KAROSA a.s." target="Obsah" onmouseover="statbar('WWW stránky podniku
KAROSA a.s. ');return true" onmouseout="statbar(' ');return true">
```

```
<area shape="default" nohref>
```

```
</map>
```

```
</body>
```

```
</html>
```


Příloha B/3

Zdrojový kód souboru Formular3.htm

```
<html>
<head>
  <title>Karosa - formulář</title>
  <meta charset="Windows-1250">
  <link rel="stylesheet" type="text/css" href="Style.css">
  <script language="javascript" type="text/javascript" src="Skripty.js">
  </script>
</head>

<body onload="start()">
<div align="center">
<br><h2>Zadejte požadavky:</h2>
U roků stejně jako u teritorií můžete vybrat více možností,<br>a sice pomocí
klávesy CTRL nebo SHIFT.<br><br>
Záleží na velikosti písmen, místo neznámých znaků použijte "%".<br>
<hr width="90%">

<form action="Karosa.asp" method="post">

<table cellspacing="2" cellpadding="2" border="0" width="80%">
<tr>
  <td align="center">Rok:<br><select name="Roky" multiple size="3"
onchange="fce(this.form.Roky.options)">
  <option value="rok1990" selected>&nbsp;1990&nbsp;
  <option value="rok1991" selected>&nbsp;1991
  <option value="rok1992" selected>&nbsp;1992
  <option value="rok1993" selected>&nbsp;1993
  <option value="rok1994" selected>&nbsp;1994
  <option value="rok1995" selected>&nbsp;1995
```

```

<option value="rok1996" selected>&nbsp; 1996
<option value="rok1997" selected>&nbsp; 1997
<option value="rok1998" selected>&nbsp; 1998
<option value="rok1999" selected>&nbsp; 1999
</select></td>
<td>Tabulky řadit dle: <select name="Dle">
<option value="Typ" id="razenityp" selected>&nbsp; Typu
<option value="ČR" id="razenicr">&nbsp; České republiky
<option value="SR" id="razenisr">&nbsp; Slovenské republiky&nbsp; 
<option value="Ostatní" id="razeniostatni">&nbsp; Ostatních
</select><br>
&nbsp; &nbsp; &nbsp; <input type="radio" name="Razeni" id="Vzestupne"
value="" checked>&nbsp; <label for="Vzestupne" accesskey="v" title="Vzestupné
řazení dle typu"><u>V</u>zestupně</label><br>
&nbsp; &nbsp; &nbsp; <input type="radio" name="Razeni" id="Sestupne"
value="desc">&nbsp; <label for="Sestupne" accesskey="e" title="Sestupné řazení
dle typu">S<u>e</u>stupně</td>
</tr>
<tr><td colspan="2">&nbsp; </td></tr>
<tr>
<td align="center" colspan="2"><input type="checkbox" name="Jensuma"
id="Jensuma" onclick="pouze_suma()" title="Tato volba zobrazí pouze celkové
sumy">&nbsp; <label for="Jensuma" accesskey="p" title="Tato volba zobrazí
pouze celkové sumy"><u>P</u>ouze suma (za typy i teritoria)</label></td>
</tr>
<tr><td colspan="2">&nbsp; </td></tr>
<tr>
<td align="center">Typ:<br><input type="text" name="Typ" id="Typ"
onkeyup="test1()"></td>
<td valign="middle"><input type="checkbox" name="Sumatyp" id="Sumatyp"
title="Zobrazí sumu za typy" checked onclick="test2()">&nbsp; <label

```

```

for="Sumatyp" accesskey="s" title="Zobrazí sumu za typy"><u>S</u>uma za
jednotlivé typy (ze rok)</label></td>
</tr>
<tr><td colspan="2">&nbsp;</td></tr>
<tr>
  <td align="center">Teritorium:<br><select name="Teritorium" multiple size="3"
onchange="test3()">
  <option value="ČR" id="cr" selected>&nbsp;<nbsp;<nbsp;<Česká republika
  <option value="SR" id="sr" selected>&nbsp;<nbsp;<nbsp;<Slovenská republika&nbsp;<nbsp;<
  <option value="Ostatní" id="ostatni" selected>&nbsp;<nbsp;<nbsp;<Ostatní&nbsp;<nbsp;<
  </select></td>
  <td valign="middle"><input type="checkbox" name="Sumater" id="Sumater"
title="Zobrazí sumu za teritoria" checked onclick="test4()">&nbsp;<label
for="Sumater" accesskey="u" title="Zobrazí sumu za teritoria">S<u>u</u>ma za
jednotlivá teritoria (za rok)</label></td>
</tr>
<tr><td colspan="2">&nbsp;</td></tr>
<tr>
  <td width="50%" align="right"><input type="button" value="Odeslat
formulář" onmousedown="parent.Nabidka.location.href='Nabidka.htm'"
onmouseup="document.forms[0].submit()"></td>
  <td width="50%" align="left"><input type="reset" value="Všechno vymazat"
onclick="parent.Nabidka.location.href = 'Nabidka.htm'"></td>
</tr>
</table>
</form>
<br>
</div>
</body>
</html>

```

Příloha B/4

Zdrojový kód souboru Karosa.asp

```
<html>
<head>
  <title>Karosa - výsledek</title>
  <meta charset="Windows-1250">
  <link rel="stylesheet" type="text/css" href="Style.css">
  <script language="javascript" type="text/javascript" src="Skripty.js">
  </script>
</head>

<body>
<%
jensuma = Request.Form("Jensuma")
roky = Trim(Request.Form("Roky"))
typ = Trim(Request.Form("Typ"))
vypistyp = typ
tera = Trim(Request.Form("Teritorium"))
sumatyp = Request.Form("Sumatyp")
sumater = Request.Form("Sumater")
If jensuma = "on" Then
  typ = "_x_"
  vypistyp = "Žádný"
  sumatyp = "on"
  sumater = "on"
  tera = "ČR, SR, Ostatní"
End If
vypister = tera
If sumatyp = "on" Then
  tera = tera + ", Celkem"
End If
```

```

dle = Request.Form("Dle")
razeni = Request.Form("Razeni")
If razeni = "desc" Then
    vypisrazeni = "Sestupně"
Else
    vypisrazeni = "Vzestupně"
End If

rok = Split(roky, ", ", -1, 1)
ter = Split(tera, ", ", -1, 1)

Set Propojeni=Server.CreateObject("ADODB.Connection")
Propojeni.Open "DSN=Karosa;UID=sa;PWD=pwd;"

If typ = "" Then
    vypistyp = "Všechny"
End If
vypisrok = ""
For i = LBound(rok) To UBound(rok)
    vypisrok = vypisrok + Right(rok(i),2) + " "
Next
If sumater = "on" Then
    vypisumater = "Ano"
Else
    vypisumater = "Ne"
End If
If sumatyp = "on" Then
    vypisumatyp = "Ano"
Else

```

```
vypissumatyp = "Ne"
```

```
End If
```

```
%>
```

```
<div align="center"><a name="top">
```

```
<table width="95%" border="0">
```

```
<tr>
```

```
<td width="10%">&nbsp;</td>
```

```
<td width="" align="left"><h2><u>Výsledky:</u></h2></td>
```

```
<td align="right" valign="top" width="33%"><small><b><u>Vaše  
požadavky:</u></b><br><b>Typ: </b><%= vypistyp %><br><b>Rok: </b><%=  
Trim(vypisrok) %><br><b>Řazení:</b> <%= dle %>, <%= vypisrazeni  
%><br><b>Teritorium:</b> <%= vypister %><br><b>Celkem typy: </b><%=  
vypissumatyp %><br><b>Celkem teritoria: </b><%= vypissumater  
%></small></td>
```

```
</tr>
```

```
</table><br></b>
```

```
<%
```

```
For i = LBound(rok) To UBound(rok)
```

```
dotaz="SELECT * FROM "&rok(i)&" WHERE ("&rok(i)&".Typ LIKE "'&typ&'")  
AND ("&rok(i)&".Typ NOT LIKE 'Celkem') ORDER BY "&rok(i)&".&dle&"  
&razeni&";"
```

```
dotaz_suma="SELECT * FROM "&rok(i)&" WHERE ("&rok(i)&".Typ LIKE 'Celkem')  
ORDER BY "&rok(i)&".Typ ;"
```

```
Set vysledek=Propojeni.Execute(dotaz)
```

```
Set vysledek_suma=Propojeni.Execute(dotaz_suma)
```

```
If vysledek.EOF AND jensuma<>"on" Then %>
```

```
Požadavek na <%= Left(rok(i),3) %> <b><%= Right(rok(i),4) %></b>: Nebyl  
nalezen odpovídající záznam.<br><br>
```

```
<% Else %>
```

```
<table border="3" cellpadding="3" cellspacing="3">
```

```
<caption align="center"><big><b>Prodej autobusů Karosa za <%= Left(rok(i),3)  
%>&nbsp;<%= Right(rok(i),4) %></b></big></caption>
```

```
<tr>
```

```
<th width="120"><span class="red">&nbsp;<span></span></th>
```

```
<% For j = LBound(ter) To UBound(ter) %>
```

```
<th width="80"><span class="blue"><%=ter(j)%></span></th>
```

```
<% Next %>
```

```
</tr>
```

```
<% Do Until vysledek.EOF %>
```

```
<tr>
```

```
<td><i>&nbsp;<%= vysledek("Typ") %></i></td>
```

```
<% For k = LBound(ter) To UBound(ter)
```

```
    If k = UBound(ter) And sumatyp = "on" Then %>
```

```
        <td align="center"><b><%= vysledek(ter(k)) %></b></td>
```

```
    <% Else %>
```

```
        <td align="center"><%= vysledek(ter(k)) %></td>
```

```
    <% End If %>
```

```
<% Next %>
```

```
</tr>
```

```
<% vysledek.MoveNext %>
```

```
<% Loop %>
```

```

<% If Sumater = "on" Then %>
<tr>
  <td><b>&nbsp;<%= vysledek_suma("Typ") %></b></td>
  <% For k = LBound(ter) To UBound(ter) %>
    <td align="center"><b><%= vysledek_suma(ter(k)) %></b></td>
  <% Next %>
</tr>
<% End If %>

</table><br><br>

<% End If %>

<% Next %>

<% Propojeni.close %>

<br><br>
[ <a href="#top" title="Na začátek stránky" onmouseover="statbar('Na začátek
stránky');return true" onmouseout="statbar(' ');return true">Na začátek stránky</a>
]<br><br>
[ <a href="javascript: history.back(1)" title="Zpět"
onmouseover="statbar('Zpět');return true" onmouseout="statbar(' ');return
true">Zpět</a> ]
<br><br><br><br>
</div>
</body>
</html>

```


Zdrojový kód souboru Typy.asp

```
<html>
<head>
  <title>Přehled typů autobusů Karosa a.s. Vysoké Mýto</title>
  <meta charset="Windows-1250">
  <link rel="stylesheet" type="text/css" href="Style.css">
  <script language="javascript" type="text/javascript" src="Skripty.js">
  </script>
</head>

<body>
<div align="center"><br>
<h2>Přehled dostupných<br>typů autobusů</h2>
Roků můžete vybrat více, a sice pomocí klávesy CTRL nebo SHIFT.
<hr width="90%">

<table width="40%" border="0" cellpadding="3" cellspacing="13">
<tr valign="top">
  <td rowspan="3" align="center"><form action="Typy.asp" method="post">
  Rok:<br><select name="Roky" multiple size="3">
  <option value="rok1990" selected>&nbsp;1990&nbsp;
  <option value="rok1991" selected>&nbsp;1991
  <option value="rok1992" selected>&nbsp;1992
  <option value="rok1993" selected>&nbsp;1993
  <option value="rok1994" selected>&nbsp;1994
  <option value="rok1995" selected>&nbsp;1995
  <option value="rok1996" selected>&nbsp;1996
  <option value="rok1997" selected>&nbsp;1997
  <option value="rok1998" selected>&nbsp;1998
  <option value="rok1999" selected>&nbsp;1999
```

```

        </select></td>
        <td><input type="reset" value="Vymazat"></td>
</tr>
<tr>
        <td><input type="submit" value="Zobrazit "></form></td>
</tr>
</table>
<br><br>

```

```

<%
Set Propojeni=Server.CreateObject("ADODB.Connection")
Propojeni.Open "DSN=Karosa;UID=sa;PWD=pwd;"

roky = Request.Form("Roky")
rok = Split(roky, ", ", -1, 1)

For i = LBound(rok) To UBound(rok)

dotaz="SELECT Typ FROM "&rok(i)&" WHERE "&rok(i)&".Typ NOT LIKE 'Celkem'
ORDER BY "&rok(i)&".Typ ;"

Set vysledek=Propojeni.Execute(dotaz) %>

<table border="3" cellpadding="3">
<tr>
        <th align="center" colspan="5"><span class="red">&nbsp;<%= Left(rok(i),3)
%>&nbsp;<%= Right(rok(i),4) %></span></th>
</tr>

<% Do Until vysledek.EOF %>
<tr>
<% For j=1 to 5 %>

```

```

    <% If vysledek.EOF Then %>
        <td width="90">&nbsp;</td>
    <% Else %>
        <td width="90">&nbsp;<%= vysledek("Typ") %></td>
        <% vysledek.MoveNext %>
    <% End If %>
<% Next %>
</tr>

<% Loop %>

</tr>
</table><br><br>

<% Next %>

<% Propojeni.close %>

[ <a href="Formular.htm" title="Zpět k formuláři" onmouseover="statbar('Zpět k
formuláři');return true" onmouseout="statbar(' ');return true">Formulář</a> ]
<br><br><br>
</div>
</body>
</html>

```

Příloha B/6

Zdrojový kód souboru Typy2.asp

```
<html>
<head>
  <title>Přehled typů autobusů Karosa a.s. Vysoké Mýto</title>
  <meta charset="Windows-1250">
  <link rel="stylesheet" type="text/css" href="Style.css">
  <script language="javascript" type="text/javascript" src="Skripty.js">
  </script>
</head>

<body background="Obrazky/Nabidka.gif" onload="statbar(' ')>
<div align="center">

<%

roky = Request.QueryString("Roky")
rok = Split(roky, ",", -1, 1)

Set Propojeni=Server.CreateObject("ADODB.Connection")
Propojeni.Open "DSN=Karosa;UID=sa;PWD=pwd;"

For i = LBound(rok) To UBound(rok)

dotaz="SELECT Typ FROM "&rok(i)&" WHERE "&rok(i)&".Typ NOT LIKE 'Celkem'
ORDER BY "&rok(i)&".Typ ;"

Set vysledek=Propojeni.Execute(dotaz) %>
<br>

<table border="1" cellpadding="1" cellspacing="1" width="90%">
```

```

<tr>
    <th align="center"><span class="red"><small><%= Left(rok(i),3)
%>&nbsp;<%= Right(rok(i),4) %></small></span></th>
</tr>

<% Do Until vysledek.EOF %>
<tr>
    <td><small>&nbsp;<a href="javascript: //aaa" title="Vybrat typ <%=
vysledek("Typ") %>" onmouseover="statbar("Vybrat typ <%= vysledek("Typ")
%>");return true" onmouseout="statbar(' ');return true"
onclick="parent.Obsah.document.forms[0].Typ.value = '<%= vysledek("Typ") %>';
test5()"><%= vysledek("Typ") %></a>&nbsp;</small></td>
</tr>
<% vysledek.MoveNext %>

<% Loop %>

</table><small>
[ <a href="Nabidka.htm" title="Zobrazit nabídku" onmouseover="statbar('Zobrazit
nabídku');return true" onmouseout="statbar(' ');return true">Nabídka</a>
]<br></small>
<% Next %>

<% Propojeni.close %>

<br>
</div>
</body>
</html>

```

Příloha B/7

Zdrojový kód souboru Styly.css

```
body {  
    color : black;  
    font-family : Garamond, MS Sans Serif, Arial;  
}
```

```
a:link {  
    color : #0033ff;  
    font-weight : bold;  
    text-decoration : none;  
}
```

```
a:visited {  
    color : #660000;  
    font-weight : bold;  
    text-decoration : none;  
}
```

```
a:hover {  
    color : #ff0000;  
    text-decoration : underline;  
    font-weight : bold;  
}
```

```
h1, h2, h3, h4, h5, h6 {  
    font-family : Helvetica, MS sans-serif, Arial;  
    font-weight : bold;  
    letter-spacing : 3px;  
}
```

```
h1 {  
    font-family : Arial Black;  
    font-size : 250%;  
    color : #66cc99;  
}
```

```
h2 {  
    font-size : 150%;  
    color : #66cc66;  
}
```

```
.red {  
    color : red;  
}
```

```
.blue {  
    color : blue;  
}
```

```
.green {  
    color : green;  
}
```

Příloha B/8

Zdrojový kód souboru Skripty.js

```
function statbar(txt) {
```

```
    window.status = txt;
```

```
}
```

```
function otevrit(url) {
```

```
    myWin = window.open (url,"displayWindow","width=400,height=420,  
status=no,toolbar=no,menubar=no,resizable=no,scrollbars=no,directories=no,  
titlebar=no,dependent=yes");
```

```
}
```

```
function zavrit() {
```

```
    close();
```

```
}
```

```
function skryti() {
```

```
    document.forms[0].Jensuma[1].checked = false;
```

```
    document.forms[0].Typ.disabled = true;
```

```
    document.forms[0].Sumater.disabled = true;
```

```
    document.forms[0].Sumatyp.disabled = true;
```

```
    document.forms[0].Teritorium.disabled = true;
```

```
}
```

```
function odkryti() {
```

```
    document.forms[0].Jensuma[1].checked = true;
```

```
    document.forms[0].Typ.disabled = false;
```

```
    document.forms[0].Sumater.disabled = false;
```

```
    document.forms[0].Sumatyp.disabled = false;
```

```
    document.forms[0].Teritorium.disabled = false;
```

```
}
```



```

function zobrazit(Formular) {
    url = (document.Formular.Select.options[document.Formular.
Select.selectedIndex].value);
    parent.Bottom.location.href = url;
}

```

```

function fce(obj) {
    var roky = "";
    for (i = 0; i < obj.length; i++) {
        if ((obj[i].selected) ||
            (obj[i].checked)) {
            roky = roky + obj[i].value + ",";
        }
    }
    n = roky.length;
    n = n - 1;
    roky = roky.substr(0,n);
    url = "Typy2.asp?Roky=" + roky;
    parent.Nabidka.location.href = url;
}

```

```

var d = new Array();
function start() {
    if (document.forms[0].Jensuma.checked) {
        a = "";
    }
    else {
        a = document.forms[0].Typ.value;
    }
    if (document.forms[0].Sumater.checked) {

```

```

        b = true;
    }
    else {
        b = false;
    }
    if (document.forms[0].Sumatyp.checked) {
        c = true;
    }
    else {
        c = false;
    }
    for (var i = 0; i < 3; i++) {
        if (document.forms[0].Teritorium.options[i].selected) {
            d[i] = true;
        }
        else {
            d[i] = false;
        }
    }
}

```

```

document.forms[0].Jensuma.checked = false;
document.forms[0].Typ.value = a;
document.forms[0].Sumater.checked = b;
document.forms[0].Sumatyp.checked = c;
document.forms[0].Teritorium.options[0].selected = d[0];
document.forms[0].Teritorium.options[1].selected = d[1];
document.forms[0].Teritorium.options[2].selected = d[2];

```

```

}

```

```

var d = new Array();
function pouze_suma() {
    if (document.forms[0].Jensuma.checked) {

```

```

        n = 1;
    }
    else {
        n = 2;
    }
    if (n == 1) {
        a = document.forms[0].Typ.value;
        if (document.forms[0].Sumater.checked) {
            b = true;
        }
        else {
            b = false;
        }
        if (document.forms[0].Sumatyp.checked) {
            c = true;
        }
        else {
            c = false;
        }
        for (var i = 0; i < 3; i++) {
            if (document.forms[0].Teritorium.options[i].selected) {
                d[i] = true;
            }
            else {
                d[i] = false;
            }
        }
        document.forms[0].Typ.value = "Žádný";
        document.forms[0].Sumater.checked = true;
        document.forms[0].Sumatyp.checked = true;
        document.forms[0].Teritorium.options[0].selected = true;
        document.forms[0].Teritorium.options[1].selected = true;
    }

```

```

        document.forms[0].Teritorium.options[2].selected = true;
    }
    else {
        document.forms[0].Typ.value = a;
        document.forms[0].Sumater.checked = b;
        document.forms[0].Sumatyp.checked = c;
        document.forms[0].Teritorium.options[0].selected = d[0];
        document.forms[0].Teritorium.options[1].selected = d[1];
        document.forms[0].Teritorium.options[2].selected = d[2];
    }
}

```

```

function test1() {
    if (document.forms[0].Jensuma.checked) {
        document.forms[0].Typ.value = "Žádný";
        alert("Nejprve musíte odznačit položku 'Pouze suma!'");
    }
}

```

```

function test2() {
    if (document.forms[0].Jensuma.checked) {
        document.forms[0].Sumatyp.checked = true;
        alert("Nejprve musíte odznačit položku 'Pouze suma!'");
    }
}

```

```

function test3() {
    if (document.forms[0].Jensuma.checked) {
        document.forms[0].Teritorium.options[0].selected = true;
        document.forms[0].Teritorium.options[1].selected = true;
        document.forms[0].Teritorium.options[2].selected = true;
        alert("Nejprve musíte odznačit položku 'Pouze suma!'");
    }
}

```

```
    }  
}  
  
function test4() {  
    if (document.forms[0].Jensuma.checked) {  
        document.forms[0].Sumater.checked = true;  
        alert("Nejprve musíte odznačit položku 'Pouze suma!'");  
    }  
}  
  
function test5() {  
    if (parent.Obsah.document.forms[0].Jensuma.checked) {  
        parent.Obsah.document.forms[0].Typ.value = "Žádný";  
        alert("Nejprve musíte odznačit položku 'Pouze suma!'");  
    }  
}
```

Příloha C

Údaje pro knihovnickou databázi

Název bakalářské práce	Prezentace dat z formátu MS Office do prostředí WWW s použitím technologií ASP a ADO
Autor práce	Tomáš Myšík
Obor	Ekonomicko-správní
Rok obhajoby	2000
Vedoucí práce	Ing. Jiří Petráš, CSc.
Anotace	Publikování databáze MS Excel do prostředí WWW pomocí technologií MS ASP a ADO
Klíčová slova	ASP, ADO, WWW, CGI, Databáze