

```

for (var j=0;j<2;j++)
{
mycurrent_row=document.createElement("TR");
for (var i=0;i<2;i++)
{
mycurrent_cell=document.createElement("TD");
currenttext=document.createTextNode("cell is row "+j+", column "+i);
mycurrent_cell.appendChild(currenttext);
mycurrent_row.appendChild(mycurrent_cell);
}
mytbody.appendChild(mycurrent_row);
}
mytable.appendChild(mytbody);
mybody.appendChild(mytable);
mytable.setAttribute("border","2");
}

function createTable()
{
// E4X
var doc = XML(document);
var mytbody = doc..body.TABLE.TBODY;
for (var j=0;j<2;j++)
{
mytbody.TR[j] = "";
for (var i=0;i<2;i++)
mytbody.TR[j].TD[i] = "cell is row " + j + ", column " + i;
}
doc..body.TABLE.#border = 2;
}

```

Záver

Význam spracovania údajov na strane klienta bude zrejme narastať. K tomu smerujú aj aktivity WHATWG – Web Hypertext Application Technology Working Group. Posledné výsledky práce tejto skupiny možno nájsť na adrese <http://www.whatwg.org/specs/web-apps/current-work/>. Odporúča sa tam robiť komunikáciu medzi klientom a serverom pri dynamickom dopĺňaní obsahu stránky s využitím XMLHttpRequest. Tejto problematike sa budeme venovať v nasledujúcom článku.

■ IMPRICH BURANSKY

S Caché v štúdiu XI.

Technológia XML sa teší čoraz väčšej obľube. Pozrime sa bližšie na to, ako jednu z oblastí XML – schémy XML – rieši databázová platforma Caché.

Zjednodušene povedané, schéma XML predstavuje slovník, ktorý firmy používajú na vyjadrenie svojich obchodných pravidiel na výmenu dát. Tieto schémy teda možno nájsť napríklad tam, kde treba zaistiť dorozumenie sa medzi subjektmi prevádzkujúcimi elektronickú komerciu. Schémy pritom môžu, ale nemusia byť štandardizované. Iba od konkrétnych obchodných partnerov potom závisí, či použijú vlastné schémy XML alebo využijú niektorý z dostupných štandardov (eXML a pod.).

Platforma Caché umožňuje pracovať s dátami ako s dokumentmi XML obojstranne, teda nie iba tieto dokumenty z objektov vytvárať, ale inštancie objektov aj načítať z dokumentov XML.

Každá trieda vytvorená v Caché a implementujúca rozhranie %XML.Adaptor je automaticky schopná pomocou metódy XMLSchema vygenerovať zodpovedajúci dokument XML opisujúci jej schému XML. Vytvorenie schémy XML pre triedu je teda pomerne jednoduchá a automatizovaná záležitosť. Programová realizácia môže vyzeráť napríklad takto:

```

//Vytvorí XML schému tried v
//baličku User a program si
//sám zistí, ktoré triedy sú
//spojené s triedou Osoba, a vytvorí
//schému aj pre ne (Adresa)

XMLSchema(file)
{
open file:"WNS" use file
//hlavička
write "<?xml version='1.0'
encoding='UTF-8'?">";
write "<s:schema
xmlns:s='http://www.w3.org/2001/XMLSchema'>";
set sc=#clas(User.Osoba).XMLSchema()
set sc=#clas(User.Auto).XMLSchema()
//Koniec
write "</s:schema>";
close file
quit sc
}

```

Opačný proces, t. j. vytvorenie definícií tried na základe schémy XML, už nemožno zautomatizovať tak jednoducho. V priebehu prekladu nových tried je totiž nevyhnutné určiť ich typ, teda či ide o triedy perzistentné, vnorené, abstraktné, registrované a pod. Tieto informácie pritom schémy XML nepodporujú. Proces načítania schémy XML preto musí byť riadený napríklad prostredníctvom prepracovaného sprievodcu. Pri použití sprievodcu je celý postup rozdelený do niekoľkých jednoduchých krokov – od zadania umiestnenia dokumentov obsahujúcich schému XML cez kontrolu schémy XML až po vytvorenie a preloženie jednotlivých tried.