

PRODUKTY TRENDY TECHNOLOGIE



průmyslová počítačová technika ... jednotky vstupů a výstupů
software pro řízení průmyslových procesů ... MES ... SQL
USB ... EIB ... TCP/IP ... HTTP ... DHTML



Control Web jako výkonný podnikový WWW server

Dlouho připravovaný systém **Control Web 5** je již na trhu od začátku letošního roku, průmyslový počítačový systém **DataLab** byl včetně nových vstupně/výstupních jednotek připojovaných přes rozhraní USB představen na jarním veletrhu Pragoregula. Skoro by se mohlo zdát, že nyní tu již moc novinek být nemůže. Vývoj se však nezastavil ani na chvíli a tak i v tomto relativně klidnějším období vzniklo několik nových zajímavých produktů a několik rozšíření a vylepšení produktů stávajících. Za bližší zmínku určitě stojí:

- **DataLab IF/EIB** - ke standardnímu procesoru s rozhraním USB je k dispozici modul pro připojení sběrnice EIB, samozřejmě jsou ovladače pro **Control Web** zdarma
- Univerzální OPC server, který může spolupracovat s jakýmkoliv zařízením prostřednictvím patřičného ovladače pro systém **Control Web**
- ActiveX komponenta pro komunikaci s jednotkami **DataLab**
- Plně dynamický WWW server v prostředí systému **Control Web** včetně komfortního administrativního rozhraní

To jsou jen některé z nových produktů, které vám mohou usnadnit vaši práci. Tento zpravodaj tedy nebude monotematický jako předchozí čísla, ale bude se zabývat několika oblastmi současných informačních a automatizačních technologií. Jako vždy jsme usilovali, aby nebyl pouhým jedním z mnoha reklamních letáků, které se dostanou na váš stůl, ale aby přinesl také nějaké užitečné a obecně použitelné informace z našeho překotně se rozvíjejícího oboru.

Systému Control Web se otevřel svět sběrnice EIB



Systémová instalační sběrnice EIB (European Installation Bus) se začíná velmi rychle rozšiřovat a není divu — s její pomocí lze velmi snadno integrovat všechny požadované funkce budov (a nejen jich) do ucelených, přehledných a velmi snadno udržovatelných systémů.

Počátky této sběrnice sahají až k roku 1991, kdy ji v rané podobě společnost Siemens vyvinula specializací svého průmyslového komunikačního protokolu Profibus. Vzniklo konsorcium EIBA (European Installation Bus Association), do kterého se postupně zapojily stovky společností. Přibližně od roku 2002 získává sběrnice EIB navíc užitečný přesah, kdy pod střešou konsorcia EIB-KNX dochází ke standardizaci a spojení s dalšími řídicími systémy budov.

EIB je velmi silně standardizována, počínaje vnitřním chováním základních mikroprocesorových stavebních modulů (BAU, BCU), přes přesnou specifikaci přenosu dat (EIB communication stack) a možných datových typů (EIS = EIB Internetworking Standard) až k abstraktnímu logickému uspořádání parametrů a konfigurací jednotek sběrnice. Tato standardizace umožňuje výrobcům části sběrnice rychle a snadno vyrábět a uživatelům sítě EIB unifikovaně a snadno sestavovat a konfigurovat.

Moderní technologie počítačové vizualizace

Zajímáte se o počítačovou grafiku? pak čtěte na str. 10

Několik otázek k systému Control Web

Odpovědi na některé otázky k tomuto programovému systému naleznete na str. 8

Že je WWW server jednou z klíčových komponent systému **Control Web** je celkem obecně známo, ale že by tento server, který vždy dostanete „v ceně“ jako součást systému mohl fungovat jako



hlavní podnikový WWW server? Ihned se nabízejí otázky, jak je na tom tento server s výkonem, bezpečností, spolehlivostí a s možnostmi programování?

více na str. 4

Univerzální OPC server

Dostupnost OPC serveru je pro každé zařízení jednou z mála záruk jeho schopnosti spolupracovat v rámci otevřeného prostředí současných informačních a řídicích systémů.

Chcete si snadno vytvořit OPC server k jakémukoliv zařízení, pro které máte ovladač pro Control Web?

O novém produktu, který vám to umožní se více dozvíte na str. 12

Novinky, zlepšení, rozšíření

Přemýšlíte, jestli si nainstalujete nový opravný balíček? Popis toho, co dostanete zdarma navíc by vám mohl usnadnit rozhodování.

Některé z novinek a rozšíření jsou popsány na str. 14

System DataLab se rozšiřuje a vyrábá

Základní myšlenka, ze které vycházejí kvality průmyslového počítačového systému DataLab, je docela jednoduchá - postupné splývání informačních a automatizačních systémů je neodvratné a je velká škoda zříkat se v průmyslové automatizaci spolehlivosti, kvality a především skvělých schopností moderních informačních technologií.



Proto dodáváme průmyslové počítače rozměry a spolehlivostí se vyrovnávající běžným PLC a současně výkonem, komunikačními schopnostmi a především možnostmi a snadností programování PLC výrazně překonávající. Samozřejmostí je operační systém Windows XP Embedded, který je určen pro počítače bez rotujícího pevného disku. Veškerý software, tj. operační systém, runtime prostředí Control Web a vlastní aplikace je na jedné paměťové kartě standardu Compact Flash.

Proto dodáváme vstupně/výstupní jednotky připojované přes rychlou sběrnici USB. Periferní systém na této sběrnici poskytuje systému odezvy v řádu jednotek milisekund. Koneckonců i vstupně/výstupní moduly uvnitř většiny PLC komunikují po nějaké (často výrazně pomalejší než USB) sériové sběrnici.

Ale rychlost a přenosový výkon nejsou ani zdaleka jedinými výhodami USB periférií. Možná ještě důležitější je plná podpora principů plug-and-play a odtud plynoucí maximální možná snadnost používání. Podstatná část kvality systému DataLab nespočívá ani tak na straně hardware (i když je postaven na nejmodernějších současných komponentách), jako spíše v oblasti programového vybavení. Právě software je tím, co rozhoduje o většině vlastností produktů.

Dnes již jde více o standardy programových rozhraní než o konkrétní komunikační protokoly. Díky standardům rozhraní programových komponent mohou spolupracovat hardwarové

i softwarové produkty různých výrobců. A zákazníkovi pak nemusí hrozit, že se stane rukojmím jednoho (nejmenovaného) dodavatele, jehož uzavřený systém s okolním světem nekomunikuje, veškeré protokoly jsou tajné a pro jistotu ani OPC server nebo HTTP server není k dispozici.

K jednotkám DataLab je k dispozici kromě již tradičních ovladačů pro systém Control Web také ActiveX komponenta. A nový univerzální OPC server také může s jednotkami komunikovat přes standardní zdarma dodávané ovladače.

Do nabídky vstupně/výstupních modulů přibyl modul analogových výstupů, poskytující standardní napěťové i proudové výstupní signály.

Novou roli pro jednotky DataLab reprezentuje modul rozhraní pro sběrnici EIB. Na jedné straně máme ověřenou a výkonnou sběrnici USB s bohatou programovou podporou a na straně druhé je k dispozici veškerá funkčnost sběrnice EIB. Zachovány jsou všechny principy plug-and-play. Všechna zařízení na EIB sběrnici lze snadno napojit na systém Control Web, na OPC server nebo i na vlastní program např. ve Visual Basicu.



Systemu Control Web se otevřel svět sběrnice EIB

EIB (European Intallation Bus) je instalační sběrnice určená především pro nasazení v inteligentních budovách, kdy s pomocí EIB lze budovu nejen řídit, ale i udržovat komfortní přehled o jejím chodu. Ve spojení s vizualizačním a řídicím systémem Control Web se možnosti využití EIB výrazně rozšiřují — všechny operace jako archivace, sledování a další úkony, které se s čistou EIB obtížně implementují, vyřeší Control Web.

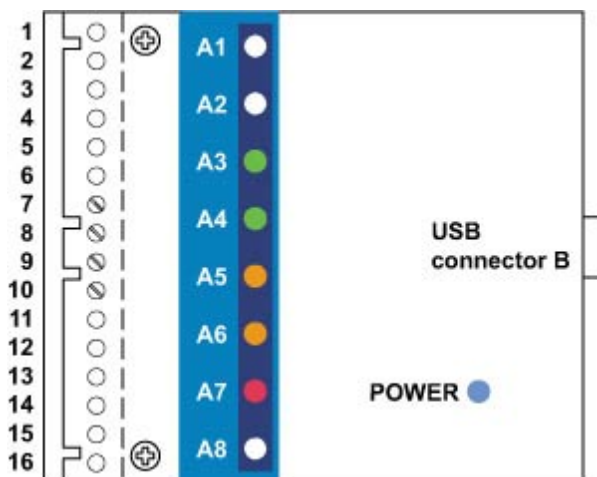
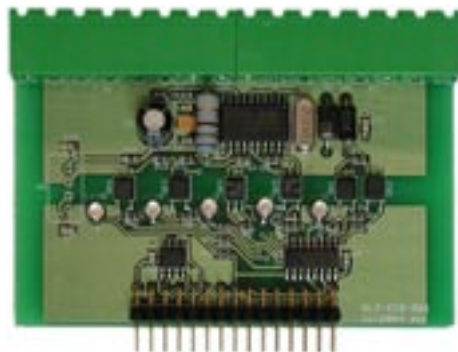
Podpora pro EIB je v systému Control Web rozdělena do dvou částí:

Hardware - jednotka rozhraní

Pro připojení EIB k počítači se systémem Control Web slouží jednotka systému DataLab IF/EIB vybavená modulem rozhraní sběrnice EIB. Jednotka se dodává jako ucelené řešení, které se jednou stranou připojuje k USB a druhou přímo k EIB sběrnici. Modul zabezpečuje:

- komunikaci po EIB sběrnici
- nezávislé napájení na stranách USB i EIB sběrnice
- galvanické oddělení USB a EIB sběrnice
- indikaci stavu EIB sběrnice i komunikačního adaptéru

Modul je vybaven průchozí konektorovou svorkovnicí pro snadné připojení. Parametry modulu splňují podmínky pro EIB zařízení připojené k médium TP (kroucená dvojlinka).



Software - ovladač pro Control Web a ActiveX komponenta

Pro připojení aplikace v systému Control Web k rozhraní DataLab IF/EIB slouží ovladač DataLab IF/EIB. Ovladač řeší standardní úkony nutné pro správný běh aplikace systému Control Web. Navíc ovladač poskytuje vlastnosti a operace běžné v prostředí EIB (např. definice komunikačních objektů s více skupinovými adresami).

Z pohledu informačních technologií lze EIB rozdělit na dvě části. První z nich je **distribuovaný systém** jednotek, kdy každá část systému (jednotka, aktor, zařízení, modul, ...) je plně samostatná a schopná vykonávat určitou činnost. Tomuto dílu odpovídají části EIB jako přenosové médium (TP = kroucený dvoudrát, PL = rozvod NN 230 V či iETS = přenos pomocí UDP), komunikace, konfigurace komunikačního zásobníku, chování jednotek při výpadku napájení apod. Druhý díl EIB je mnohem zajímavější — obsahuje skupiny a jejich vazby.

Skupiny a jejich vazby jsou totiž to hlavní, co EIB dává její obrovskou jednoduhost a eleganci.

Skupina je vysoce abstraktní, nicméně docela jednoduchý pojem. Skupina má datový typ (je-li to číslo či logický údaj), typu odpovídající hodnotu (velikost teploty) a adresu. To je vše. Skupina tak může nést veličiny, vnitřní stavy sběrnice, případně může sloužit jen jako procesní nástroj (třeba zvýšení jasu), vždy se nachází jakoby „nad“ sítí jednotek. Do skupiny může být (pomocí adresy skupiny) zapojena každá jednotka sítě (přesněji její jeden či více komunikačních bodů = SAP = Service Access Point). Každá jednotka sítě tak získává svůj obraz skupiny, který si pamatuje a se kterým operuje. Jedna abstraktní skupina takto existuje ve všech jednotkách v konkrétní podobě. Celá EIB potom řeší pomocí komunikací jednu jedinou věc — jak zajistit, aby tyto konkrétní obrazy skupin v jednotkách byly shodné, aby si jejich hodnoty navzájem odpovídaly.

Ovladač DataLab IF/EIB lze použít v systémech Control Web 5 a Control Web 2000 vždy v prostředí operačních systémů Windows NT v5.0 a vyšší, což odpovídá současným operačním systémům Windows 2000, Windows XP a Windows 2003.

Upozornění: Jiné operační systémy, jako například Windows ME nebo Windows NT 4.0, neobsahují systémová volání nutná pro běh ovladače, a ovladač v nich proto není schopen pracovat. Vzhledem k požadavkům na robustnost průmyslových aplikací je použití bezpečných systémů s jádrem Windows NT v5.0 v každém případě nanejvýš žádoucí a tak podpora pouze systémů Windows 2000 a Windows XP v praxi nepředstavuje omezení.

Ovladač komunikuje s rozhraním DataLab IF/EIB pomocí USB, což znamená:

- ovladač nepotřebuje žádnou konfiguraci portů, komunikační rychlosti ani adresování jednotek,
- ovladač pro svou práci vyžaduje, aby operační systém „znal“ rozhraní DataLab IF/EIB. To je zajištěno pomocí speciálního systémového USB ovladače, který je nutné do operačního systému nainstalovat. Po instalaci se tento systémový ovladač stane vnitřní součástí operačního systému (podobně jako např. ovladač grafické karty).

EIB skupiny

EIB skupina je označena adresou. Adresa skupiny (skupinová adresa, group address) je 15bitové číslo, což dává možnost definice až 32 768 různých skupin.

Skupinové adresy se zapisují dvojím způsobem (je to věcí volby tvůrce EIB nasazení) jako čísla oddělená lomítky:

- pomocí dvou čísel, hlavní skupiny (main group) a podskupiny (subgroup), přičemž hlavní skupina může být v rozsahu 0–15 a podskupina v rozsahu 0–2047. Zápis skupinové adresy v této podobě může vypadat například takto: 4/1225.
- pomocí třech čísel, hlavní skupiny (main group), střední skupiny (middle group) a podskupiny (subgroup), přičemž hlavní skupina může být v rozsahu 0–15, střední skupina v rozsahu 0–7 a podskupina v rozsahu 0–255. Skupinová adresa v této podobě může vypadat například takto: 2/2/173.

Ovladač DataLab IF/EIB samozřejmě s adresami skupin pracuje a podporuje jejich oba možné zápisy. V parametrickém souboru ovladač přirozeně používá uvedeného zápisu s lomítky, v mapovacím souboru (většinou automaticky vytvořeném) a v aplikaci systému Control Web však zápisu adres s lomítky nelze použít.

Datové typy

Kromě adresy je se skupinou spojen i její **datový typ**. Datový typ skupiny, jako části systému EIB, je přesně dán EIB

standardem (EIS) a v ovladači se proto definuje právě jako tento EIB (EIS) typ. Control Web disponuje řadou různých typů, takže provázání s EIS typy je přirozené

Rychlost EIB

Nejčastěji používané (a tedy základní) přenosové médium pro EIB je kroucený dvoudrát (TP), pomocí něj se k EIB připojuje i rozhraní DataLab IF/EIB. Toto přenosové médium používá přenosovou rychlost 9 600 bps, což při započtení délky nejkratšího možného EIB TP telegramu (8 byte), prodlevy a potvrzení telegramu (ACK) dává komunikační rychlost 64 EIB telegramů za sekundu.

To není nijak moc, pokud bychom chtěli komunikovat bezhlavě. Pro technologie budov to však bohatě stačí, protože rychlosti sledovaných a řízených dějů jsou v řádech desítek a stovek sekund.



Control Web jako obecný vizualizační a řídicí nástroj pracuje svým tempem a vzhledem k jeho mohutnému výkonu je velmi snadné neopatrnou (či špatnou) aplikací EIB sběrnici zcela zahltit. Velmi proto doporučujeme dodržet v aplikaci EIS standard, z něhož část vyjímáme:

Upozornění: EIBA Handbook Series, Internetworking standards, 2.3 General Requirements for Interworking, 2.3.1 Busload: Rychlost opakování — rychlost opakování by měla být volena velmi obezřetně, neboť značně ovlivňuje provoz na sběrnici a hrozí zde riziko zahlcení. Předpokládaná zátěž sběrnice by měla být již součástí plánu instalace EIB sítě.

Doporučují se následující pravidla:

- pro uživatelské (ruční) operace by vzdálenost mezi odesílanými telegramy měla být nejméně 200 milisekund,
- pro automatické operace by vzdálenost mezi odesílanými telegramy měla být v řádech sekund či minut.

Poznámka: Ve většině případů automatických operací je postačující rychlost opakování v rádech minut — například měření teploty, jasů, času apod. vždy měří veličiny, které se mění pomalu.

Tip: Instalace z adresáře na CD-ROM není nezbytná. Oba soubory je možné okopírovat na jakékoliv médium. Pokud například cílový počítač nemá mechaniku CD-ROM, je možné ovladač okopírovat na pevný disk nebo na USB Flash Disk a instalovat jej z této kopie.

Principy dynamických webových aplikací

Control Web jako podnikový WWW server

HTTP server zabudovaný v systému Control Web nabízí mnohem více než pouhou vizualizaci a řízení technologických procesů prostřednictvím WWW prohlížečů. Control Web poskytuje vše co potřebujete k vývoji a provozování moderní rozsáhlé WWW aplikace.

Zeptáte-li se dnešních studentů oborů informačních technologií, co si představí pod pojmem počítačový „program“ či „aplikace“, zřejmě jen málokdo pod tímto pojmem dnes rozumí jediný spustitelný soubor, ze kterého se zavede kód procesu operačního systému. S propojením počítačů do celosvětové sítě, s příchodem mobilních zařízení jako PDA či „chytřích“ mobilních telefonů, se standardizační komunikačních protokolů a aplikačních vrstev se význam pojmu aplikace posunuje. Dnes se „aplikací“ rozumí spíše sada komponent a skriptů pracujících na straně serveru i v klientských prohlížečích, komunikujících s databázovým strojem a prostřednictvím HTTP serveru zprostředkovávající informace klientům. Podobu WWW aplikace tak získávají služby jako například hledání v telefonním seznamu, nalezení místa na mapě, vyhledání autobusových či vlakových spojů apod. Tento koncept aplikací se začíná uplatňovat i mimo prostředí rozlehlého Internetu. Veškeré existující nástroje, znalosti a zkušenosti programátorů lze uplatnit i pro aplikace pracující v prostředí lokálních sítí (intranetů) a stejnou podobu tam může získat třeba účetní systém či skladová agenda firmy. Dokonce i na lokálním počítači existuje řada aplikací, které pro vytvoření uživatelského rozhraní používají stejného konceptu, tedy řadu HTML stránek. S dalším rozvojem různých oborů informačních technologií, jako jsou bezdrátové sítě, přenosné počítače s nízkou spotřebou apod., se tento trend bude dále posilovat.

Aplikace počítačů v průmyslu vykazují často až podivuhodnou rezistenci vůči přílivu nejnovějších technologií a otevřených standardů (tato rezistence je často dobře odůvodněna požadavky na maximální spolehlivost a robustnost, na druhé straně stejně často je hlavní příčina v neochotě naučit se cokoliv nového i pokud nová technologie výrazně zvýší užitečnost aplikace a zlevní řešení). Přesto výhodnost internetových a intranetových řešení pro celou řadu aplikací je natolik zřejmá, že začínají pronikat i do tohoto konzervativního odvětví. Internetový prohlížeč (WWW browser) je dnes přítomen na každém počítači a stejně tak každý počítač v podnikové

sféře je zapojen do sítě. Možnost zpřístupnit vizualizaci a případně i řízení průmyslového procesu z libovolného počítače se tak pro zákazníky stává velkým lákadlem.

Samozřejmě nelze od vizualizace v prostředí WWW prohlížeče očekávat stejné možnosti jako od aplikačního programu pracujícího na lokálním počítači. V prostředí prohlížeče jsme omezeni internetovými standardy formátů dokumentů a obrázků, možnostmi programování (skriptování), standardy pro zabezpečení apod. Další omezení se týkají například reakčních dob systému na požadavky na komunikaci s PLC nebo měřicími jednotkami apod. Zatímco lokálně běžící aplikace musí být schopna řídit komunikaci v reálném čase a případně reagovat na zpoždění či poruchy komunikace, realizovat stejnou funkci (se srovnatelnou dobou odezvy) v prostředí WWW prohlížeče je téměř nemožné. Přitom platí jedno pravidlo - čím širší má být okruh klientů schopných k aplikaci přistoupit (například pomocí různých typů prohlížečů pracujících na různých operačních systémech), tím více omezení bude na aplikaci kladeno. Extrémně může být například aplikace pracující na miniaturních zobrazovacích mobilních telefonech, pro jejichž tvorbu musíme vystačit s elementárním HTML kódem a jediným formátem obrázků.

Přítomnost zabudovaného HTTP serveru, díky němuž mohou k aplikaci přistupovat klienti prostřednictvím WWW prohlížečů, byla jedním z důvodů pojmenování programového systému pro rychlý vývoj průmyslových vizualizačních a řídicích aplikací Control Web. Ačkoliv v první verzi systému Control Web 3 byl HTTP server vhodný spíše pro menší aplikace v rámci intranetu, narůstající požadavky zákazníků vedly k podstatnému rozšíření funkčnosti, usnadnění programového řízení, zpřístupnění hlaviček HTTP protokolu, zvyšování robustnosti (například detekci a odražení pokusů o získání kontroly nad serverem útokem typu „buffer overrun“) apod. Všechny tyto inovace vedly k verzi HTTP serveru v systému Control Web 5 schopné plnohodnotně pracovat nejen jako WWW brána do technologického procesu, ale také v roli podnikového WWW serveru s kompletním redakčním systémem, schopným obsloužit velké množství klientů. Protože statické, „ručně“ tvořené HTML stránky již neodvratně patří minulosti, samozřejmě na tuto roli nestačí HTTP server sám, ale potřebuje k tomu další komponenty systému Control Web, především pro přístup

Ovladač se dodává k jednotkám rozhraní DataLab IF/EIB zdarma.



k SQL databázím. Klíčovou roli při tvorbě takových webových aplikací sehraje výkonný skriptovací jazyk systému Control Web.

Generování HTML stránek

Přístup k WWW stránkám je dnes již považován za naprostý základ počítačové gramotnosti - v žádném kurzu používání počítače se s WWW prohlížečem nelze minout. Povědomí, co se za WWW službou skrývá, ale už zdaleka není běžné, i když samotný princip je velmi jednoduchý.

Aby si dva počítače mohly vyměňovat data, musí oba používat stejný dorozumívací jazyk. Protokol užívaný při komunikaci mezi klientem a serverem služby WWW nese označení HTTP (z anglického Hyper-Text Transfer Protocol). V prostředí Internetu jsou používány i další protokoly, například SMTP pro přenos elektronické pošty, FTP pro přenos souborů apod. Nedejme se ale zmýlit názvem protokolu, HTTP lze použít k přenosu jakýchkoliv dat, nejen hypertextů. Pomocí HTTP lze přenášet obrázky, spustitelné binární soubory, komprimované archivní soubory apod. Protože server služby WWW komunikuje s klienty pomocí HTTP protokolu, bývá také nazýván HTTP server.



Když pomineme technické detaily navazování síťového spojení na portu služby HTTP serveru, celé kouzlo WWW spočívá v odesílání HTTP požadavků z klientské aplikace k serveru, na které server odesílá HTTP odpovědi. Základním požadavkem je požadavek typu GET - získej data. O jaká data má klient zájem určuje tzv. URL (z anglického Universal Resource Locator). URL má v prostředí WWW podobnou funkci jako jméno

souboru na lokálním počítači - pojmenovává zdroj dat. URL ale navíc může nést další informace a parametry pro server.

Jak s daty naložit je už záležitost klientské aplikace (WWW prohlížeče). Pokud server poskytne HTML soubor (textový soubor formátovaný podle pravidel HTML - Hyper-Text Markup Language), prohlížeč jej zformátuje podle značek v souboru obsažených (značky určují např. nadpisy, dělení odstavců apod.). HTML stránka ale může obsahovat další data, např. obrázky. Obrázky ale nejsou obsaženy přímo v textu HTML souboru, ale ve stránce je na ně uveden pouze odkaz v podobě URL. Klient pak pošle další dotaz serveru s danou URL a server odpoví blokem dat, tentokrát obsahujícím požadovaný obrázek. Vrácený obrázek pak klient zobrazí v rámci stránky.

Podobnost URL se jménem souboru svádí k představě, že HTTP server pouze převede URL obdrženou od klienta na jméno lokálního souboru a tento soubor klientovi vrátí. V případě statické WWW aplikace je tomu skutečně tak - všechny HTML dokumenty, obrázky a další soubory tvořící aplikaci jsou připraveny na disku a HTTP server je čte a odesílá klientům. Nevýhoda takové aplikace je ale velmi pracná údržba. Jakákoliv změna v řadě dokumentů provázaných hyper-textovými odkazy je obtížná a zdlouhavá. Žádná rozsáhlejší WWW aplikace takto konstruována není prostě z důvodu naprosté neudržovatelnosti. Řešení je zřejmé - na místo statické struktury souborů na straně serveru běží aplikace, která obsahy stránek vytváří algoritmicky.

Jako příklad můžeme uvážit úvodní stránku WWW aplikace, která má obsahovat odkazy na články vložené za poslední týden. Ruční údržba takové stránky by představovala neustálé sledování nově vložených článků a dopisování odkazů na ně do HTML souboru představujícího úvodní stránku, a také odmazávání starších odkazů. Naproti tomu u stránky tvořené kódem procedury probíhá vše automaticky. Procedura se dotáže SQL serveru na články s datem vložení nejpozději 7 dní pozpátku a algoritmicky vytvoří HTML dokument, který odešle klientovi. Klientská aplikace samozřejmě nemůže rozlišit, jak HTML dokument, který jí poslal server vznikl a nakonec to ani nepotřebuje. Jestliže dokument odpovídá pravidlům HTML, může jej korektně zobrazit. To je celé kouzlo dynamické WWW aplikace.



Pro ilustraci uvedme elementární příklad dynamické tvorby kořenové stránky v systému Control Web 5:

```
httpd WebServer;
pages
  item
    path = '/';
    call = GenerateIndex();
  end_item;
end_pages;

procedure GenerateIndex();
begin
  PutText( '<html><head><title>Ukázka</title></head>' );
  PutText( '<body>Stránka generovaná dynamicky</body></html>' );
end_procedure;
end_httpd;
```

Pokud WWW prohlížeč odešle serveru dotaz s URL ,/' (základní nebo tzv. indexová stránka), server ví, že tento soubor nemá hledat na disku, ale k jeho vytvoření má zavolat proceduru GenerateIndex. Kód této procedury vytvoří text stránky postupným voláním PutText. WWW prohlížeč obdrží HTML dokument:

```
<html><head><title>Ukázka</title></head>
<body>Stránka generovaná dynamicky</body></html>
```

a jak bylo řečeno dříve, není schopen rozlišit, zda tento dokument byl uložen v podobě souboru na disku nebo zda byl vytvořen algoritmicky.

V případě jednoduché stránky nepřináší dynamické generování žádné zjevné výhody, snad mimo skutečnost, že celá WWW aplikace může pracovat zcela bez přístupu na disk. Uvažme ale následující proceduru GenerateIndex:

```
procedure GenerateIndex();
var
  i : integer;
begin
  access_count = access_count + 1;
  PutText( '<html><head><title>Dynamická stránka</title></head><body>' );
  if display_table then
    PutText( '<table border="1" width="30%" align="center">' );
    for i = 1 to lines do
      PutText( '<tr><td>řádek </td><td> ' + str( i, 10 ) +
        ' </td></tr>' );
    end; (* for *)
    PutText( '</table>' );
  else
    PutText( '<center><ul>' );
    for i = 1 to lines do
      PutText( '<li>řádek ' + str( i, 10 ) + '</li>' );
    end; (* for *)
    PutText( '</ul></center>' );
  end;
  PutText( '<hr><b>Stránka vygenerována ' + date.TodayToString() +
    ', v ' + date.NowToString() + '.<br>Počet přístupů: ' +
    str( access_count, 10 ) + '</b>' );
  PutText( '</body></html>' );
end_procedure;
```

V tomto případě je výhoda dynamického generování již zjevná. Především jediná procedura tvoří HTML text odpovídající tabulce nebo seznamu na základě nějaké podmínky. Dále stojí za zdůraznění, že délka stránky není předem dána, ale záleží na hodnotě proměnné lines. Na samém konci stránky je zobrazeno počítadlo přístupů, aktuální datum a čas. Každý klient vždy přečte ze serveru HTML dokument s jiným textem (lišící se minimálně zobrazeným počtem přístupů a také datem a časem), i když požádá o stejnou stránku.

Aplikace nebo jen prohlížení dokumentů?

Prapůvod služby WWW leží v systému zpřístupňujícím vědcům dokumenty v evropském středisku jaderného výzkumu CERN. Možnost zabudovat do textu odkaz na jiný dokument (odtud název hyper-text) dělá z obyčejných textů jednoduchou aplikaci, reagující na přání uživatele - kliknutí na odkaz způsobí zavedení nového dokumentu do prohlížeče. Pokud by možnosti HTML zabudováním odkazů na jiné soubory do textu končily, asi bychom nemohli hovořit o

aplikačním prostředí. Popularita WWW ale zapříčinila prudký vývoj tohoto standardu a postupně začaly přibývat možnosti nejen zabudování obrázků do dokumentu, přesnějšího formátování, ale také možnosti tvorby formulářů, v nichž uživatelé mohou zadávat data pro aplikaci. Vývoj se ale nezastavil a 0160HTML standard obsahuje možnost psaní skriptů (skripty v HTML stránce představují programový kód, který je ve WWW prohlížeči vykonáván), kaskádní styly, dynamické HTML apod. Dnes tedy už můžeme hovořit o HTML jako o relativně bohatém a mocném aplikačním prostředí.

Součástí definice protokolu HTTP je mimo požadavků čtení dat ze serveru (metodou GET) i zápis dat na server metodou PUT. V praxi se ale této metody nepoužívá, protože není podporována klientskými aplikacemi a především obsahuje řadu bezpečnostních rizik - v principu je nemyslitelné, aby klienti ukládali soubory na server na zadané URL. Proto je odesílání dat od klienta na server omezeno na dvě metody protokolu HTTP:

- Lze použít metody GET s daty zakódovanými v URL.
- Přímou k odesílání dat na serveru je určena metoda POST.

Poznamenejme, že HTTP protokol nedefinuje mechanismy zpracování dat. Záleží výhradně na serverové aplikaci, jak s daty naloží.

Ačkoliv je metoda POST původně navržena pro odesílání dat z HTML formulářů, byla rozšířena i o možnost odesílání celých souborů. Na rozdíl od metody PUT ale v tomto případě URL neurčuje kam daný soubor na serveru uložit, ale spíše definuje, která část serverové aplikace bude soubor zpracovávat. Jak se souborem naložit je opět záležitostí implementace serverové aplikace - např. může uložit soubor do databáze apod.

Systém Control Web nabízí řadu způsobů zpracování dat odeslaných uživateli na server.

- Nejjednodušším způsobem je definování vazeb mezi jmény ovládacích prvků v HTML stránce a datovými elementy v aplikaci. Pokud uživatel odešle data z formuláře do aplikace, patřičné datové elementy nabudou hodnot vyplněných ve formuláři.
- Pokud je část nebo celá HTML stránka tvořena kódem procedury, může programátor pomocí procedury GetURLData získat řetězec, který obsahuje jména a hodnoty ovládacích prvků z formuláře. Je pak na programátorovi, aby z tohoto řetězce získal patřičné hodnoty.
- Protože GetURLData lze volat jen z procedury reagující na požadavek GET, data zasílaná metodou POST lze zachycovat procedurou OnFormData. Tato procedura je volána vždy, když serveru přijdou data z formuláře, ať již metodou GET nebo POST.
- Pokud aplikace využívá rozšíření metody POST dovolující uživatelům odesílat na server celé soubory, může využít událostní procedury OnPostFile.

Zde můžeme konečně vysvětlit odkud se



vzaly hodnoty proměnných `display_table` a `lines` v předchozím příkladu. HTML formulář obsahuje ovládací prvky pojmenované a pokud je formulář odeslán serveru, jména a hodnoty těchto ovládacích prvků jsou připojeny do URL:

```
http://localhost/
default.htm?display_format=true&-
line_count=10
```

V HTTP serveru pak stačí uvést mapování mezi jmény ovládacích prvků a jmény proměnných v aplikaci:

```
httpd WebServer;
static
lines : integer;
display_table : boolean;
access_count : longint;
end_static;

forms
item
id = 'line_count';
output = lines;
end_item;
item
id = 'display_format';
output = display_table;
end_item;
end_forms;
...
end_httpd;
```

Problémy může způsobovat nejednoznačnost definice chování serveru při odpovědi na metodu POST v rámci standardu HTTP. Součástí POST je totiž URL, definovaná atributem ACTION v definici formuláře v HTML dokumentu. Dle definice tato URL ale slouží k identifikaci entity, již se data odesílají v rámci POST týkají. Není specifikováno, zda-li mají být data odkazované URL v POST vrácena (stejně jako po GET) či nikoliv. Praktické zkušenosti ukázaly, že optimální chování serveru po POST, maximálně ulehčující tvorbu WWW aplikací, je chování odpovídající metodě GET. Existují-li data odkazovaná v URL u POST, jsou vrácena s kódem 200 OK, pokud data ale neexistují, není vráceno 404 Not Found jako po GET, ale 204 No Content.

Optimalizace datového toku po TCP/IP síti

Mechanismy vyrovnávacích pamětí, kdy jsou data dočasně uchovávána v místě potřeby a nikoliv pokaždé přenášena z trvalého uložení, se ukázaly jako mocný způsob zrychlení a zefektivnění práce v řadě technických i programových systémů. Kupříkladu vyrovnávací paměť v procesoru mu umožní pracovat mnohonásobně rychleji než by mu umožnila rychlost pamětí, vyrovnávací paměť disku zvýší jeho efektivní přenosovou rychlost apod.

Stejným způsobem mohou vyrovnávací paměti HTTP protokolu výrazně urychlit zpřístupnění WWW stránek. WWW stránky obsahují řadu statických obrázků, které se po dlouhou dobu nemění a je zbytečné

aby je prohlížeč pokaždé ze serveru zaváděl. Z tohoto důvodu si každý prohlížeč uschovává určité množství dokumentů a obrázků na lokálním disku počítače, odkud je může zavést a zobrazit nepoměrně rychleji než po seberyčlejší IP síti.

V rámci Internetu i intranetu se lze setkat se specializovanými servery pracujícími jako vyrovnávací paměť HTTP protokolu. Pokud více klientů (např. v rámci podnikové sítě) nepřistupuje na vzdálený server přímo, ale prostřednictvím tzv. proxy-serveru, mohou tyto klienti výrazně zrychlit přístup k WWW stránkám. První klient způsobí zavedení stránek do vyrovnávacího serveru, v případě dalších klientů se vyrovnávací sever pouze ubezpečí, zda data na původním serveru nebyla změněna a pokud ne, vrátí data ze své vyrovnávací paměti na místo jejich pomalého přenosu přes Internet. Oba mechanismy (vyrovnávací paměť WWW prohlížeče i specializované vyrovnávací servery) jsou si velmi podobné a využívají velmi podobných mechanismů.

Klíčová otázka v každém systému s vyrovnávací pamětí je zajištění konzistence dat. Pokud se např. obrázek na serveru změní, bylo by od klienta chybné zobrazovat obrázek uložený v lokální vyrovnávací paměti. Ovšem klient nemá jinou možnost, jak zjistit aktuálnost své vyrovnávací paměti, než dotázat se serveru. S každým blokem dat přenášeným HTTP protokolem (pod blokem dat rozumíme např. HTML dokument, obrázek apod.) je přenášena informace o okamžiku poslední modifikace. Klient tedy ví, jak starý je dokument uložený v lokální vyrovnávací paměti, musí jen zjistit stáří právě aktuálního dokumentu na serveru.

K tomu byla do HTTP protokolu zabudována další metoda nazvaná HEAD. Tato metoda odpovídá metodě GET (obsahuje URL i další údaje v požadavku), ovšem klient očekává jako odpověď pouze hlavičku bloku dat. Protože hlavička obsahuje informace o čase poslední modifikace, může se klient rozhodnout, zda o data požádá tentokrát metodou GET nebo zda použije data z lokální vyrovnávací paměti.

Použití dotazu HEAD může ušetřit zbytečné přenosy, pokud ale dokument není aktuální, znamená prodloužení přenosu o jeden dotaz HEAD a odpověď. Z tohoto důvodu bývá v prostředí Internetu používána alternativní metoda. Klient požádá o data server metodou GET, ale do protokolu uvede hlavičku informující server aby data odeslal jen pokud byla modifikována od zadaného data odpovídajícího datu poslední modifikace dat v lokální vyrovnávací paměti. Server sám posoudí aktuálnost dat a pokud jsou k dispozici data novější, přímo odpoví. Pokud ne, informuje klienta, že data nebyla modifikována a žádá data neodesílat. Tento způsob zaručuje skutečně minimální zatížení sítě a optimální přenos dat. HTTP server v systému Control Web podporuje oba způsoby optimalizace přenosu.

Výše popsané mechanismy si lze velmi snadno představit u statických dokumentů uložených jako soubory. Okamžik poslední modifikace každého souboru je zapsán v souborovém systému a HTTP server jej může využít. Pokud je ale dokument generován dynamicky, situace se velmi komplikuje:

- U dynamického generovaného dokumentu je okamžik poslední modifikace vždy nastaven na aktuální čas. Klient tedy nikdy nemůže mít aktuální verzi a vždy musí přenést data ze serveru.
- Ne vždy se ale dynamicky generovaná stránka skutečně liší od stránky generované předchozím dotazem. Pokud např. algoritmus slouží pouze k dekodování URL a vrácení dat uložených v databázi, je zbytečné nastavovat okamžik poslední modifikace na aktuální datum a čas. Pomocí procedury `SetLastModified` může klient nastavit okamžik poslední modifikace a Control Web HTTP server automaticky vrátí data klientovi nebo mu oznámí že data nebyla modifikována. Ačkoliv u WWW serverů nebývá limitní výkon počítače, ale kapacita přenosových linek, procedura generující stránku může voláním `GetHeader` zjistit hodnotu hlavičky `If-Modified-Since` a rozhodnout se, zda-li je zapotřebí stránku vygenerovat či ne. Pokud voláním `SetLastModified` nastaví datum rovné (nebo menší než) datu v hlavičce `If-Modified-Since`, server klientovi odpoví kódem 304 Not Modified a tak není nutno ztrácet čas vytvářením stránky.
- V případě, že algoritmus generující stránku pouze přeměruje datový tok do souboru voláním `RedirectToFile`, jako okamžik poslední modifikace také nebude použit okamžitý čas, ale čas poslední modifikace souboru.

Control Web dokáže dynamicky generovat dokumenty zcela bez zásahu uživatelského programu. Pokud kupříkladu chceme WWW prohlížečům přenášet okamžitou podobu nějakého virtuálního přístroje, zajistíme to velmi jednoduchým mapováním URL obrázku na viditelný virtuální přístroj v aplikaci:

```

httpd WebServer;
instruments
  item
    path = '/img1';
    instrument = panel_energie;
  end_item;
  item
    path = '/img2';
    instrument = panel_voda;
  end_item;
  ...
end_instruments;
...
end_httpd;

```

Kdykoliv se v HTML stránce objeví odkaz na obrázek „img1“, server nebude prohledávat disk (soubor s tímto jménem by ani nenalezl), ale vykreslí okamžitou podobu přístroje se jménem `panel_energie`. V tomto případě je okamžik poslední modifikace vždy nastaven na aktuální čas.

Mocnost HTTP serveru zabudovaného do systému Control Web 5 dokazuje aplikace vyvinutá pro běh na serveru <http://www.mii.cz>. Aplikace obsahuje nejen „klientskou“ část, která zobrazuje data návštěvníkům serveru (čtete-li tento článek na serveru www.mii.cz, data pro Vás připravil Control Web), ale také administrativní část umožňující pohodlnou administraci celého serveru.

Serverová aplikace odpovídá všem nárokům na moderní informační WWW server:

- Veškeré texty jsou ukládány ve formátu XML užívajícím jednotnou definici typu dokumentů. Tím je zaručeno jednotné a konzistentní formátování v rámci všech stránek. Změnou XML transformací lze najednou změnit vzhled všech textů. Server tak zcela odděluje obsah dat a jejich formátování.
- Žádná data nejsou uložena staticky. Veškeré texty jsou generovány do HTML formátu ze zdrojových XML souborů dynamicky za běhu aplikace.
- Veškerá data jsou uložena v SQL databázi. Datový obsah tak lze jednoduše zálohovat či replikovat.
- Vzhled stránek je definován algoritmicke na základě struktury dat. Pokud např. do nabídky firmy přibude nový produkt, stačí přidat do aplikace jeho popis a zařadit jej do příslušné kategorie. Anotace s odkazy na detailní popis se automaticky objeví v produktové stránce, případně ve stránce novinek apod.
- Zcela automatizovaná je podpora vládání obrázků do dokumentů. Index obrázků je uložen v databázi a tak potenciální výměna obrázků je snadnou záležitostí. V rámci transformace

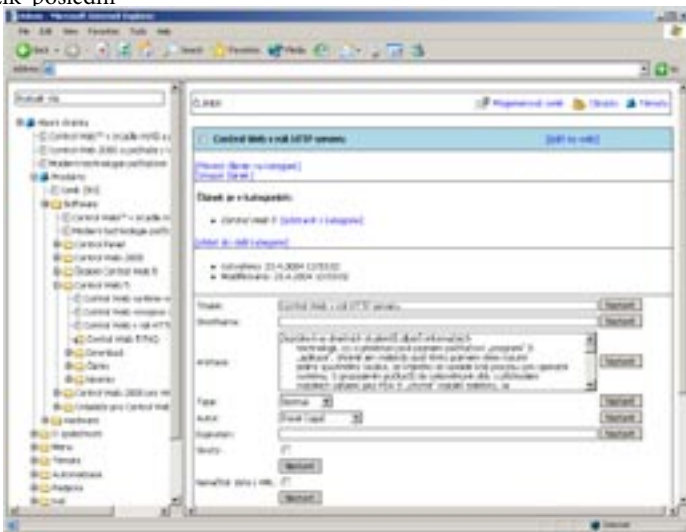
XML do HTML jsou podle atributů s obrázkem spojených automaticky generovány malé či velké náhledy, odkazy na obrázek v plném rozlišení apod.

Tvůrci obsahu mají k dispozici administrativní rozhraní, umožňující měnit strukturu kategorií a přidávat či modifikovat články, popisy, obrázky apod. Celé rozhraní pracuje v rámci standardního WWW prohlížeče, z důvodů bezpečnosti lze přístup k tomuto rozhraní omezit nejen jménem či heslem, ale také určitými IP adresami či maskami IP sítí.

Administrátorské rozhraní pro tvůrce obsahu WWW stránek

Základní operace s kategoriemi (přepojování podkategorií, editace anotace apod.) jsou implementovány pomocí HTML formulářů. Nicméně editační možnosti ovládacích prvků HTML formulářů jsou natolik omezené, že komfortní editace delších článků se složitějším formátováním je prakticky nemožná. Proto systém nabízí stažení zdrojového tvaru článků v XML formátu na lokální počítač, kde je možné článek editovat jakýmkoliv XML editorem. Poté je možné XML opět nahrát na server. Serverová aplikace zkontroluje správnost odkazů a dostupnost obrázků a případně vyzve uživatele k zadání umístění obrázků na lokálním počítači, automaticky je přenesou na server a zařadí do databáze.

Aplikace <http://www.mii.cz/> demonsturuje pouze malou část schopností systému Control Web zaměřenou na tvorbu distribuovaných aplikací v prostředí Internetu a intranetů pro klienty používající WWW prohlížeče. Velké množství dalších oblastí, jako např. tvorba distribuovaných aplikací založená na „tlustých klientech“, sdílených a synchronizovaných datových sekcích, automatické generování DHTML aplikací atd. ale mnohonásobně přesahuje možnosti tohoto článku.



Několik otázek k systému *Control Web*[®]

Je tomu již přes dvanáct let, kdy byl na trh uveden první systém *Control Panel 1*. Tak se začala odvíjet historie až po současný nejmodernější produkt *Control Web 5*. První *Control Panel* byl ve své době mimořádně úspěšným produktem. Přinesl novou a velmi silnou koncepci skládání aplikačních programů z nezávislých komponent, reálný čas a programovatelnost. Vždyť v tehdejší době veškeré stávající SCADA programy byly pouhými cyklickými smyčkami čtení tagů s parametrizovatelnou grafikou (a konekcí mnohé systémy jsou takové dodnes). Tehdejší úspěch systému *Control Panel* dokazuje i několik tisíc licencí prodaných ve velmi krátkém čase. Intenzivní vývoj pokračuje 12 let a mezi původním produktem a současným systémem *Control Web* je rozdíl asi jako mezi papírovým drakem a moderním Boeingem. Dodnes mezi odbornou veřejností přžívá řada „mýtů a pověr“. Některé vychází z před mnoha lety platných faktů, některé z konzervatismu a předsudků často se vyskytujících v oboru průmyslové automatizace a řada z nich je pouhou iracionální ideologií. Člověk je asi více emocionální než racionální bytostí, jen se podívejme kolem sebe, kolik lidí hodnotí současné operační systémy od Microsoftu podle vlastností (a k nim vázaných pověstí) prvních systémů *Windows 95* a zcela bez vztahu k realitě a kvalitě dnešních systémů *Windows 2000* a *Windows XP*. Možná nebude na škodu si v souvislosti s často tradovanými (a to dokonce i v odborném tisku či v akademickém prostředí) nejasnostmi uvést pár faktů. Pokusme se alespoň pár „otázek na tělo“ blíže objasnit...

Je *Control Web* standardní?

Dnes se již našťastí podobnými „problémy“ téměř nikdo netrápí, uživatelé současných informačních a komunikačních technologií jsou moudřejší, reálně posuzují vlastnosti produktů a neztrácejí čas malichernostmi, jako je např. velikosti ikonky a jejich správná barva. Nicméně začněme na úvod tímto dříve velmi častým, ale bohužel dosti neurčitým dotazem. Námitka vůči standardnosti jde téměř vždy ruku v ruce s neschopností svého nositele definovat, co mají být ony dotyčné standardy a jak (a kým) jsou definovány. Softwarové technologie se trvale rychle vyvíjejí a nejsou proto postiženy normalizací v klasickém slova smyslu. Na druhé straně zde existuje spousta tzv. de facto průmyslových standardů (což není nic jiného, než ty neju-

spěšnější z původně vlastnických firemních a vojenských technologií), bez nichž by se tento průmyslový obor stal oblastí nepropojitelných a nespolečných řešení a bez kterých bychom neměli např. Internet, SQL databáze, OpenGL grafiku atd. a bez kterých bychom neměli ani samotný *Control Web*. Ten je totiž postaven

právě na takovýchto všeobecně přijatých a rozšířených technologiích. *Control Web* vždy (a po této stránce to neměl právě v oblasti průmyslové automatizace nijak lehké) prosazoval koncepci interoperability na bázi otevřených standardů a proto je v této oblasti etalonem standardnosti a kompatibility. Měl by být ve frontě adeptů na kritiku nestandardnosti na jednom z posledních míst. *Control Web* je pro tyto své vlastnosti často používán v roli spojovacího článku mezi původně nespolečnými programovými i technickými prostředky v rozsáhlých heterogenních sítích. Další oblastí, kterou by mohli mít na mysli šířitelé fám o nestandardnosti je grafické uživatelské rozhraní vývojového prostředí (nikoliv již aplikace, její rozhraní si každý definuje sám). Zde je již mluvit



o nějakém standardu vskutku velmi problematické. Jediné, co se trochu blíží kritériím pro používání tohoto termínu je řada nepsaných pravidel používaných firmou Microsoft v jejich produktech řady Microsoft Office a Microsoft Developer Studio. I zde integrované vývojové prostředí systému *Control Web* dosti úzkostlivě dodržuje takto stanovená a zažitá pravidla - a to dokonce přísněji, než je tomu u mnohých programů přímo od Microsoftu. Tyto dlouhodobě se vyskytující mýty o nestandardnosti mají pravděpodobně pozadí v již davné historii, kdy systémy *Control Panel* pracovaly ve vlastním operačním systému. To bylo v době, kdy dominoval systém *Windows 3.1* a pomalu se objevovala novinka v podobě *Windows 95*. Tehdy jsme po velmi zralé úvaze nepovažovali tato prostředí za ta pravá pro trvalý běh kritických úloh v průmyslové automatizaci (a z této doby také pochází většina argumentů proti stabilitě systémů *Windows*). *Control Panel* tehdy pracoval v chráněném módu ve víceúlohovém prostředí s virtuální pamětí a běžně dosahoval několikaleté doby mezi servisní-



mi odstavky. Jeho grafické uživatelské bylo tehdy odvozeno z unixového standardu *Motif* a nikoliv z *Windows* verze 3. To asi tehdejší propagátoři šestnáctibitových *Windows* mohli považovat za nestandardnost. S příchodem *Windows NT* a *Win32 API* se použitelnost systémů *Windows* pro průmyslovou automatizaci dramaticky změnila. Tehdy vznikla nová vývojová větev systému *Control Web*, který se stal ve své oblasti de facto průmyslovým standardem.

Je *Control Web* „čistým“ SCADA systémem?

Není. *Control Web* skutečně není tím, co někteří označují jako tzv. čistý SCADA systém. Avšak za nedostatek to může považovat skutečně jen člověk s velmi omezeným obzorem. Nároky na funkčnost tzv. vizualizačních nebo operátorských aplikací trvale rostou. Dnes již většinou nestačí pouhé jednoduché zobrazování dění v průmyslové technologii spolu s obvyklou archivací dat

a správou alarmových stavů. Tato jednoduchá funkčnost je v parametrizovatelných cyklických systémech napevno naprogramovaná a obvykle ji většina programů zvládá bez problémů. Ve většině současných nasazení operátorských dohledových systémů je zákazník požadováno mnohem více - často to bývají složité algoritmy reakcí na určité události (analýza a komplexní zpracování a prezentace dat, rekonfigurace systému, řízení technologie atd.), atypické požadavky na komunikace v heterogenním prostředí (TCP/IP, rádiové sítě, GSM, SMS, GPRS atd.), spolupráce s jinými systémy v rámci počítačové sítě (vazba na podnikové informační systémy, řídicí a monitorovací systémy reálného času i jiné SCADA systémy atd.) a v neposlední řadě pružná implementace řady internetových technologií. A to mluvíme jen o klasických SCADA aplikacích. Na druhé straně při řešení úloh v průmyslové automatizaci, laboratořích, výuce, modelování a simulaci, řízení v reálném čase, komunikaci a správě dat a ostatních oborech trvale vznikají tisíce požadavků, na které tzv. čisté SCADA programy z principu své funkce nemohou stačit. A to jsme se ani nezmiňovali o rostoucích požadavcích na rozsah platform. Se systémem Control Web můžeme v jednom prostředí vytvářet přenositelné aplikace pracující na mobilních telefonech, PDA, řídicích jednotkách strojů i na síti výkonných redundantních serverů. To je pro ochranu investic do technického i programového vybavení a lidských znalostí velmi cenné. Zde je velká hodnota systému Control Web - nemusíme totiž ihned na začátku vše umět používat a všemu rozumět, máme ale jistotu, že při jakýchkoliv předem nečekaných požadavcích zadání nás tento systém nenechá na holičkách. I když většinu možností třeba nikdy nepoužijeme, je příjemné vědět, že je máme kdykoliv k dispozici. Čistý SCADA systém totiž dnes znamená totéž jako hloupý SCADA systém.

Je spolehlivost systému Control Web závislá na implementaci aplikace?

Není, i když je aplikace vytvořena zcela „nesprávně“, spolehlivost a stabilitu systému to nijak neohrozí. Aplikace v prostředí systému Control Web pracují v mantinelech tzv. paměťově bezpečného modelu. Chybně vytvořená aplikace samozřejmě nemusí splňovat požadavky na ni kladené, ale nemůže ohrozit spolehlivost a stabilitu vlastního systému

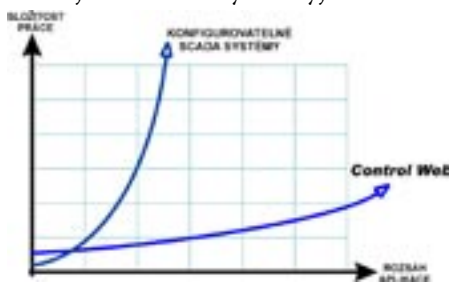
Control Web (jiným problémem pak je vůbec zamezit tomu, aby programátoři dělali chyby :-).

Lze v systému Control Web vytvořit aplikaci bez programování?

Ano, řada aplikací je takto udělána. Aplikace na úrovni funkčnosti běžných SCADA systémů lze vytvářet v grafickém prostředí zcela bez nutnosti cokoli programovat a tedy i zcela bez nutnosti umět programovat.

Je tvorba aplikací v prostředí systému Control Web pracnější než v parametrizovatelných SCADA systémech?

Pro vytvoření svým rozsahem a funkcí velmi omezených aplikací může být jejich realizace v prostředí systému Control Web o něco málo pracnější než pouhé vyplnění konfiguračního formuláře. U každé jen trochu větší a složitější aplikace se nám však potenciálně větší prvotní vklad několikanásobně vrátí v podobě dokonalého přehledu nad strukturou dat a funkcí aplikačního programu. Tato souvislost je zřetelná z grafu, kde na vodorovné ose máme rozsah aplikace a na svislé složitost vývoje a udržování. Parametrizovatelné SCADA systémy mohou mít nižší počáteční práh, křivka složitosti práce však s rostoucím rozsahem aplikace velmi rychle roste (takováto závislost je charakteristická pro čistě vizuální vývojové nástroje s omezenými vyjadřovacími



prostředky). Výrazové prostředky systému Control Web jsou bohaté a jeho křivka nárůstu složitosti je velmi plochá. Tyto skutečnosti jsou v teorii programovacích technik obecně známy a opakovaně nám je potvrzují ohlasy od uživatelů, kteří mají zkušenosti s realizací velkých projektů v různých prostředích.

Je Control Web zaměřen hlavně na menší a střední aplikace?

To rozhodně není, zde je asi trochu matoucí jeho nízká cena. Právě naopak Control Web je často řešením v situacích, kde se konvenční (tj. konfigurovatelné s cyklickým obíháním

tagů) produkty dostávají za meze svých možností. Control Web je často nasazován právě k řešení těch nejrozsáhlejších a nejsložitějších úloh, na kterých si již „čisté“ SCADA systémy vylámaly zuby. V těchto případech je nutno mít přesně pod kontrolou objemy zpracovávaných dat i metody a algoritmy jejich přenosu. Právě konvenční cyklické systémy jsou použitelné pouze pro menší aplikace, u rozsáhlých systémů se již zpravidla nevystačí s produkty, jejich často jediným způsobem vzájemné komunikace je periodická masivní replikace veškerých dat. U malých a nenáročných nasazení (tj. s malým počtem dat, bez přesného reálného času, bez složitých algoritmů zpracování dat a řízení, bez nutnosti řízení komunikace v distribuovaném prostředí a bez nutnosti provozu na více platformách) si můžeme vystačit téměř s jakýmkoliv konvenčním SCADA systémem. Pro řešení rozsáhlých a složitých systému však již nutně potřebujeme prostředky, které disponují obdobnými možnostmi jako např. systém Control Web.

Jak může být Control Web dobrý, když je tak levný?

Zde by bylo možno uvést řady příkladů, kdy levnější programové vybavení svou kvalitou výrazně překonává své dražší konkurenty. A zvláště u rozsáhlejších a v malých sériích prodávaných programových systémů se souvislost mezi kvalitou a cenou (většinou docela dobře platící u spotřebního zboží) naprosto vytrácí. Např. u ekonomických systémů může být horší systém i o dva řády dražší než systém lepší. Levnější systém bývá často nasazen ve větším počtu instalací a tak je zde větší pravděpodobnost vychtání všech chyb. Sériový výrobek je obvykle spolehlivější než kusový prototyp trvale přizpůsobovaný na míru každému svému nasazení. **Nemusí tedy být ve všech případech správné se při výběru řídit jen cenou, někdy může být výhodnější koupit si ten lepší produkt i když je levnější...**

Podle naprosté většiny ohlasů od uživatelů je Control Web považován za stabilní, spolehlivý a velmi bohatě vybavený nástroj. Control Web jistě není ani v nejnovější verzi 5 zcela dokonalým produktem a je toho stále velmi mnoho (jako ostatně u veškerého programového vybavení), což je nutno zlepšovat. Jen paradoxně právě v těch oblastech, kterých se týkají výše uvedené nejasnosti a nepřesnosti, je Control Web výrazně silnější a lepší, než je v tomto oboru běžné.

Nemíváte také někdy pocit, že průmyslová automatizace je tak trochu „nudným oborem“? Je to asi tím, že se jí lidé zabývají převážně proto, že se jí pod tlakem okolností zabývat musejí. Je to jejich práce, jsou to jejich pracovní povinnosti. Technologie používané v průmyslové automatizaci jsou proto hodně tradiční, ověřené lety a častí i desetiletými praktických nasazení. Investují-li lidé do svých koníčků, chtějí vždy to nejlepší, aktivně se zajímají o nejmodernější technologie a velmi pečlivě při každém nákupu zvažují poměr cena-výkon. V zaměstnání však většinou pracují s přidělenými nástroji a motivace k používání nejvýkonnějších prostředků zde není obvykle tak přímočará. Naštěstí ale ani v tomto oboru není vše jen černé nebo jen bílé. I zde existuje dost aktivních a tvůrčích lidí, kteří stále hledají cestu vpřed a v předstihu používají ve svých řešeních moderní rychlé sběrnice, internetové technologie a při vizualizaci např. 3D virtuální realitu ...

Ještě relativně nedávno se pro vizualizaci průmyslových procesů běžně používalo programové vybavení pracující v textovém módu grafických adaptérů. Pomocí semigrafických znaků tyto programy svižně zobrazovaly sloupcové indikátory, potrubí a otevřené či uzavřené ventily. A to bylo v době, kdy jsme již mnoho let psali dopisy a e-maily v grafickém prostředí Windows. Kde je však v dnešní době semigrafika. Její používání mohla opodstatnit jen technická nedostatečnost počítačů a existence textových terminálů. A ti, kteří tvrdili, že pro průmyslovou automatizaci to bude navždy stačit, neboť semigrafika je naprosto dostačující, naštěstí neměli pravdu (dobře si tehdejší „plodné“ diskuse dodnes pamatují). Standardem dnešní doby je při vizualizaci technologických procesů využívání dvourozměrné vektorové grafiky. A opět je jisté, že to tak navždy nevydrží.

2D vektorová a bitmapová grafika

V dalším textu se budeme věnovat grafickým technologiím přístupným v prostředí Microsoft Windows. O zobrazování dvourozměrné grafiky a textu se stará systém GDI (Graphics Device Interface). Je to jeden ze tří hlavních a nejstarších částí operačních systémů Windows. Moduly KERNEL, USER a GDI jsou základem Windows již od verze 1.0, dvourozměrná grafika je tedy již „odjakživa“ pevnou součástí služeb jádra operačního systému.

Systém GDI se s novými generacemi operačních systémů stále zdokonaluje a rozšiřuje, stále však tento grafický jazyk udržuje kontinuitu s původní, dnes již velmi letitou, koncepcí. Současné požadavky na 2D vizualizace samozřejmě předpokládají podporu animací, průhledností, gradientních výplní a obrazových transformací. Proto bývá ve vizualizačních programech obvykle nad GDI vystavěn složitější a mocnější vykreslovací systém. Dnes je i samotný systém Windows vybaven dalšími technologiemi pro vytváření obrazu. Je zde systém GDI+, jehož API je přímo tvořeno hierarchií C++ tříd a především velmi mocný systém Direct3D. GDI+ přináší mnoho nových možností ve 2D vektorové grafice (např. gradientní štetce, maticové transformace souřadného systému i barevného prostoru, alpha blending atd.), při manipulaci s bitmapami i v typografii. Těžko odhadovat budoucí podporu tohoto rozhraní, může se jednat o slepou vývojovou větev. Naproti tomu Direct3D je postaven na standardech rozhraní COM (Component

Object Model). Tento 3D vykreslovací systém lze (stejně jako OpenGL) velmi dobře využít i pro vektorovou 2D grafiku.

Jak jsme již uvedli, GDI služby jsou součástí jádra operačního systému (i když je docela těžké vysledovat, co vše je řešeno na straně klienta a co uvnitř jádra). Řada volání těchto služeb z aplikace do jádra systému představuje určitou režii spojenou s přepnutím kontextu. Tato režie je často delší, než vlastní realizace grafického primitiva grafickým akcelerátorem. Sebelepší grafická karta pak již není schopna přispět k lepšímu výkonu grafiky. Tyto potíže pomáhá řešit akumulace funkčních volání a jejich následné hromadné předání systému při dokončení kresby. Takto je 2D grafika řešena i ve vektorovém vykreslovacím systému v prostředí Control Web.



Obr.1: Vlastní kresby lze v prostředí Control Web ukládat do knihoven pro další použití

Dalším problémem bývá rušivé blikání obrazu během překreslování. Je způsobeno tím, že vykreslovací systém obvykle přímo modifikuje paměťový rámec v grafické kartě, který je využíván pro generování obrazového signálu pro monitor. Systém Control Web se s tímto vypořádává zcela principiálně a shodně pro veškerou 2D grafiku v aplikaci obsaženou. Každému panelu lze nastavením parametru mode = off_screen zajistit vykreslování do neviditelné paměti a přenesení výsledku do viditelného rámce najednou následně po dokončení kresby. Tato operace je vždy podporována v hardware grafické karty a je velmi rychlá. Výsledkem je čistý obraz bez rušivého blikání.

Obr.2: Ukázka 2D vektorového obrázku v prostředí systému Control Web

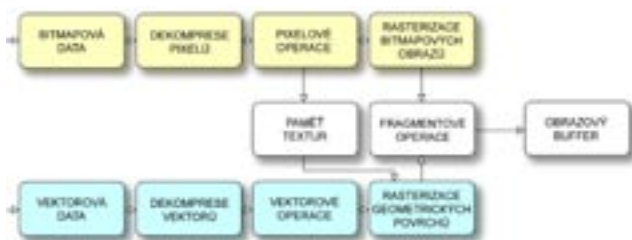


Je-li takto přenášen vždy výsledný hotový obraz, je již lhostejno, jakou technologií byl vytvořen (nezáleží ani na tom, jde-li o 2D nebo 3D grafiku). Moderní vykreslovací systémy pro generování obrazu využívají vektorový procesor pracující s plovoucí řádovou čárkou a rasterizer pro tvorbu výsledného bitmapového obrazu. Takto pracují např. vykreslovací systémy Macromedia Flash nebo Adobe PDF a Adobe SVG Viewer (Scalable Vector Graphics - W3C XML standard pro přenositelnou grafiku).

Trochu podobným směrem kráčí budoucí generace Windows známá pod jménem Longhorn. Nové grafické rozhraní Avalon - XAML (eXtensible Application Markup Language) umožní deklarativní popis grafického uživatelského rozhraní. I když pro vytváření obrazu používáme některou z moderních technologií, bez spolupráce s GDI se alespoň při manipulaci s kontextem zařízení a poté při umísťování výsledného obrazu do patřičného okna stejně neobejdeme. Systém GDI nás bude provázet ještě velmi dlouho a jistě bude součástí připravovaného OS Windows Longhorn. Nás ale nyní nejvíce zajímá, co můžeme ku prospěchu našich aplikací využívat již nyní. Pojďme tedy k tomu nejzajímavějšímu - k interaktivní 3D grafice.

3D grafika

Co pro nás dělá 3D vykreslovací systémy tak zajímavými? Pomineme-li nedostížně vizuálně působivé obrazy, pak jsou to rozhodně schopnosti současných grafických akcelerátorů. 3D akcelerátor máme v každém současném počítači a nevyužívat jej je velká škoda. Jako bychom např. trvali pouze na černobílém zobrazení v době, kdy jsou již dávno všechny monitory barevné. Princip tvorby obrazu je podobný, jako jsem již popisovali v předchozí kapitole. Základem výkonu akcelerátorů je zřetězení a paralelizace. Např. současný akcelerátor GeForce 6800 s více než 220 miliony tranzistorů má ve svém hardware 6 paralelních vektorových procesorů 16 paralelních pixelových procesorů (při zápisu do z-bufferu a stencil bufferu dokonce 32 paralelních pixelových kanálů - a to je pro Control Web a moderní vykreslovače vůbec velmi přínosné). Výsledkem jsou datové propustnosti v řádech desítek gigabyte za sekundu a schopnosti vykreslit stovky milionů stínovaných trojúhelníků za sekundu. No uznejte sami - můžeme si tyto možnosti dovolit nechat ležet ladem?



Obr.3: Zřetěžený proces tvorby obrazu systémem OpenCL

Poznámka: Datové propustnosti současných grafických karet se nám mohou na první pohled zdát až zbytečně předimenzované. Vždyť i již nikoliv nejmodernější GF5900 FX zvládne datový tok přes 27 GB/s. Pokusme se však odhadnout přibližné nároky při vykreslování scény s rozlišením 1280x1024 pixelů s frekvencí 60 snímků/s. Uvažujeme-li z-buffer a barevný buffer s 32 bity na pixel a 24-bitové RGB textury, pak při přepisu pixelu (tj. čtení i zápis do z-bufferu) s mícháním barev (blending - tj. čtení i zápis do bufferu barev) a dvěma trilineárně filtrovanými texturami, pak dostaneme 64 byte na jeden pixel (a to neuvážujeme docela častý vícenásobný přepis pixelu). Při těchto předpokladech dostaneme datový tok více než 5 GB/s. Takto vytvořený obraz by jistě nevypadal příliš realisticky, většinou musíme dále počítat s rezervami pro víceprůchodové vykreslování, zápisy a čtení stencil bufferu atd. Ve skutečnosti budeme potřebovat datové toky několikanásobně vyšší. Navíc je zde ještě značná zátěž způsobená tokem vektorových dat (150 milionů trojúhelníků za sekundu spotřebuje asi 3.3 GB/s). Je tedy zřejmé, že skvělé vlastnosti současných grafických akceleratorů „zužitkujeme“ zcela bez problémů.



Obr.4: Obraz se zrcadlením, odlesky a prostorovými stíny vykreslený systémem Control Web s využitím víceprůchodových vykreslovacích technik

Krátké ohlédnutí do historie

Oblast 3D počítačové grafiky prochází v současné době prudkým technickým rozvojem. Hnací motorem tohoto pokroku jsou bezesporu počítačové hry. Díky velikosti trhu s hrami se vyvíjejí stále výkonnější 3D akceleratori. Výkony v oblasti stovek milionů trojúhelníků za sekundu a datové toky v řádu desítek gigabyte za sekundu již patří k běžným parametrům.

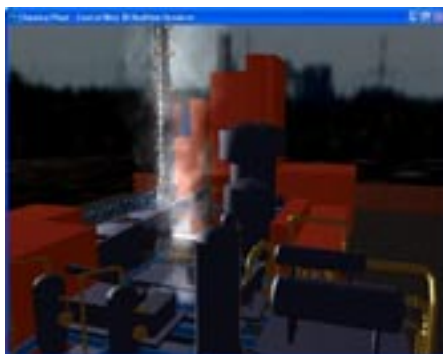
Milníkem ve vývoji počítačové 3D grafiky byl rok 1992, kdy firma Silicon Graphics (SGI)

uvedla svůj Reality Engine. Tento grafický akcelerator obsahoval 8 grafických procesorů s hardwarovou podporou transformací, osvětlení a rasterizace včetně mapování textur. Stál asi milion dolarů a byl velký jako domácí lednička. Dnes

i levné grafické karty za méně než 50 dolarů výkon tohoto ve své době revolučního zařízení výrazně překonávají. Za 10 let tak klesla cena srovnatelného grafického adaptéru asi 20000 krát!

Éru levných grafických 3D akceleratorů pro PC zahájila firma 3Dfx svou kartou Voodoo 1 v roce 1996. Tento úspěšný adaptér nenechal nikoho na pochybách o užitečnosti hardwarové podpory 3D grafiky. Dnes je již 3D grafický akcelerator tak samozřejmou součástí každého domácího PC jako je např. myš nebo pár reproduktorů. A blízká budoucnost - hardwarová podpora 3D grafiky již nezadržitelně míří i do oblasti počítačů do dlaně a mobilních telefonů.

Jak jsme se již zmiňovali, s trochou uživatelského pragmatismu můžeme říci, že nám nezáleží na principech a způsobech tvorby obrazu. Zajímá nás jen to, aby byl obraz hezký a byl vytvořen velmi rychle. Zde se budeme držet zásady, že co nelze vykreslit v reálném čase, je lépe nekreslit vůbec. A nechceme-li se specifickou problematikou 3D grafiky hlouběji zabývat, nemusíme. Při tvorbě aplikace v prostředí systému Control Web se často ani nemusíme zatěžovat budováním opravdu prostorové scény, stačí využít část potenciálu 3D vykreslovacího systému v běžné 2D vizualizaci.



Obr.5: Vizualizace průmyslové technologie, vykreslené jedním průchodem bez využití dalších grafických efektů

I tehdy nám 3D vykreslovací systém v prostředí Control Web přinese navíc mnohý užitek:

- využijeme mohutného vykreslovacího výkonu 3D akceleratoru - část grafické zátěže se přenechá do paralelně běžící GPU grafické karty
- aplikace získá perfektní vzhled osvětleného a plastického vygenerovaného obrazu
- aplikační program běží plynuleji a přesněji v reálném čase - vykreslovací server běží ve vlastním vláknu a jeho klienti běžící v reálném čase se grafikou vůbec nezabývají



Obr.6: Při jednopřechodovém vykreslování maximálně využijeme „hrubý“ výkon grafické karty a nejvíce šetříme výkonem CPU

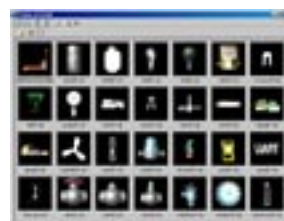
Budeme-li někdy chtít využít z této grafické technologie více, máme k dispozici:

- editaci 3D prostorové scény „na místě“ přímo v ploše integrovaného vývojového prostředí
- volný pohyb kamery ve scéně včetně interpolací pozice i úhlů, a to při editaci i za běhu aplikace
- řadu předpřipravených 3D objektů pro rychlé a bezstarostné sestavení vizualizace
- stromový systém transformací a animací objektů a možnost plně programové kontroly scény
- částicový systém pro realistické a dynamické zobrazování prostorových efektů s předem připravenými objekty (oheň, kouř, jistry atd.)
- editovatelné materiály s možností průhlednosti, texturování atd.
- import 3D modelů (*.obj, *.3ds) z různých 3D modelovacích programů
- vizualizace zdrojů světla a simulace odlesků a rozptýlů světla v optice kamery



Obr.7: Prostorové stíny současně s vizualizací zdroje světla

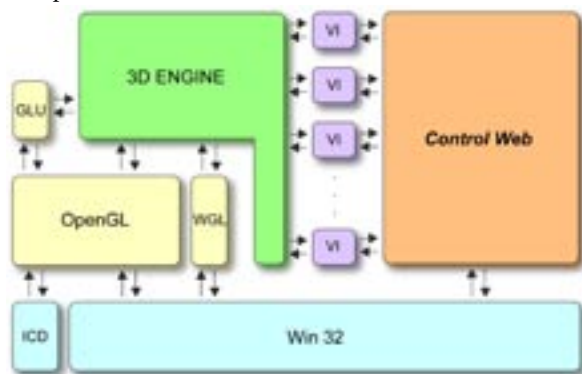
- zrcadlové plochy
- planární stíny
- prostorové dynamické stíny s výkonným a robustním algoritmem
- a mnoho dalšího ...





Obr.8: In-place 3D editor vývojového prostředí systému Control Web umožňuje volný pohyb ve vytvářené scéně včetně vykreslování všech obrazových efektů

Tyto možnosti dokumentují technologickou vyspělost vykreslovacího systému a samozřejmě jich ve svých aplikacích vždy nemusíme využívat. Ale je příjemné mít je k dispozici.



Obr.9: Architektura 3D vykreslovacího stroje v systému Control Web

Důležité je také informace o architektuře 3D vykreslovacího systému a jeho zařazení do struktur prostředí Control Web. Jak je vidět na obr. 8, vykreslovací systém poskytuje své služby pouze patřičným virtuálním přístrojům a nikoliv jádru systému Control Web. Vykreslovací systém se tedy dynamicky zavádí až spolu s prvním virtuálním přístrojem, který jej potřebuje. Rozhodně tedy nemusíme

mít obavu, že by v aplikacích soft PLC bez vizualizace nebo v aplikacích s 2D vizualizací komponenty 3D vykreslovacího systému jakkoliv překážely, zabíraly místo ve virtuálním paměťovém prostoru a jakkoliv zdržovaly start aplikace.

3D vykreslování v reálném čase

Je dobré se také zamyslet, proč bychom se vůbec v průmyslové automatizaci měli zabývat 3D grafikou (když už jsme se tak dlouho bez ní docela obešli :-)). Argument proti se dá vymyslet asi jen jeden jediný, a to ten, že sestavit vizualizaci dá o něco více práce. Do jisté míry je to pravda, avšak právě snadnost tvorby 3D vizualizací byla hlavním návrhovým cílem 3D grafického prostředí v systému Control Web. V řadě případů můžeme vytvořit 3D scény zcela bez hlubší znalosti problematiky i bez nutnosti relativně pracného kreslení externích 3D modelů. Na druhé straně argumentů ve prospěch 3D zobrazování je hned několik:

* **Vzhled** — v současnosti se jedná o nejkvalitnější dostupnou zobrazovací technologii. Vykreslování obrazu ve 3D prostoru (3D rendering) se využívá pro tvorbu filmů, v počítačových hrách, při zobrazování modelů ve vědě a výzkumu, v CAD programech, při prezentaci dat atd. je tedy nejvyšší čas i pro průmyslovou automatizaci.

* **Výkon** — 3D akcelerátor je samozřejmou součástí každého současného PC a je škoda tohoto potenciálu nevyužívat.

* **Architektura** — klient-server vykreslovací systém, maximum vykreslování je řešeno mimo jádro operačního systému. To přináší lepší možnosti optimalizace a plynulejší reálný čas aplikací.

* **Cena** — používáme stejný počítač a aplikace vytváříme prakticky stejným způsobem jako jsme byli dříve zvyklí u 2D vizualizací. Získáváme tedy vyšší kvalitu bez nutnosti cokoliv připlácet.



Obr.10: Logo systému Control Web vykreslované v reálném čase přímo systémem Control Web

Nevyužívat pro vizualizace 3D grafický systém, je-li v podobě grafické karty a instalovaných ovladačů na počítači přítomen (a to je téměř vždy), je jako bychom svým autem stále jezdili jenom na první rychlostní stupeň.

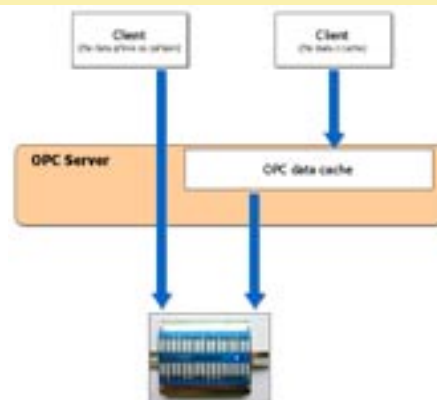
Oblast průmyslové automatizace je tradičně velmi konzervativním oborem, který bohužel ani zdaleka nepatří k technologickým průkopníkům na poli informačních technologií. Avšak v systému Control Web v trojrozměrné grafice můžeme nyní využít nejen skvělého výkonu moderních akceleratorů, ale díky implementaci řady nejnovějších softwarových technologií a algoritmů třírozměrného zobrazování v reálném čase získat pro své aplikace rychlou a vizuálně působivou grafickou prezentaci. Nemusíme již tedy, jak je často k vidění, prostorové zobrazení pouze napodobovat pomocí předem připravených bitmap. Můžeme mít opravdové trojrozměrné zobrazování v reálném čase!

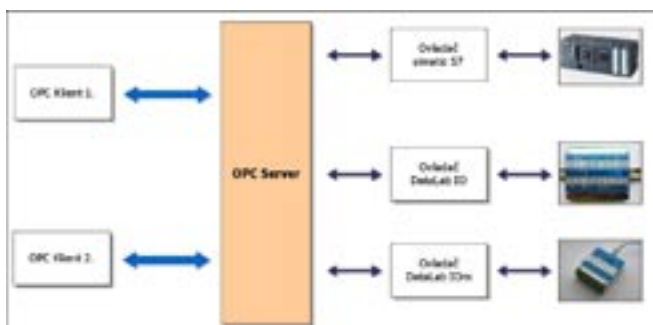
Univerzální OPC server

OLE for Process Control (OPC) představuje první úspěšnou iniciativu standardizující komunikační rozhraní mezi prvky průmyslové automatizace — průmyslovými automaty, čidly a akčními členy na jedné straně a řídicími či operátorskými počítači a průmyslovými informačními systémy na straně druhé. Ze standardu OPC profitují především uživatelé, kteří díky němu přestávají být vázáni na programové a technické vybavení podporující pouze vlastnické protokoly zavedených firem (zejména pokud tyto firmy brání své postavení právními překážkami zabraňujícími implementaci daného protokolu třetím stranám). Standard OPC spravuje nezisková organizace OPC Foundation (<http://www.opcfoundation.org/>) a implementace tohoto standardu je dostupná všem bez jakýchkoliv licenčních poplatků.

OPC tvoří programovou vrstvu mezi technickým vybavením a programy s tímto hardware komunikujícími. Prvky průmyslové automatizace vybavené OPC serverem jsou nyní podobně snadno a bezproblémově použitelné jako např. grafické karty vybavené ovladačem pro daný operační systém — nikdo není nucen zabývat se technickými detaily konstrukce čipu dané karty a jejím programovým rozhraním, stačí instalovat ovladač a vše pracuje. Stejně tak stačí na daném počítači instalovat patřičný OPC server pro použitý hardware a programy schopné se na tento server napojit (OPC klient) jsou s tímto hardware schopny komunikovat.

Technicky je standard OPC založen na komponentové technologii COM firmy Microsoft. Název sice napovídá spíše na použití nadstavby COM pro tvorbu složených dokumentů — OLE (Object Linking and Embedding), ale dnes lze jen těžko stanovit hranici kde končí COM a kde začíná OLE, zvláště když se do hry zapojí další





marketingové názvy, jako např. „Active X“. Protože sama firma Microsoft technologii COM používá k implementaci řady komponent v operačních systémech Windows, je COM široce podporován a obsahuje řadu vlastností velmi užitečných i pro OPC, jako například globální registraci komponent (každý OPC klient přesně ví, jak se dostat k OPC serveru), kategorie komponent (OPC klient může snadno zjistit, které OPC servery a jakých verzí jsou na daném počítači dostupné) apod.

Univerzální OPC server je novým produktem, který ke komunikaci se zařízením využívá standardních ovladačů systému Control Web. Pro všechna zařízení, k nimž existuje ovladač kompatibilní se systémem Control Web, existuje nyní také plnohodnotný OPC server. OPC server na jedné straně komunikuje prostřednictvím ovladače s určitým zařízením a na druhé straně nabízí plnohodnotné standardní OPC rozhraní. K jednomu OPC serveru je možné připojit libovolné množství zařízení různých typů (vždy prostřednictvím ovladače). Současně s každým OPC serverem může samozřejmě současně komunikovat libovolný počet klientů. Data z jednoho zařízení tak může sdílet více klientů. Adresní prostor všech zařízení je mapován do hierarchického adresního prostoru OPC serveru.

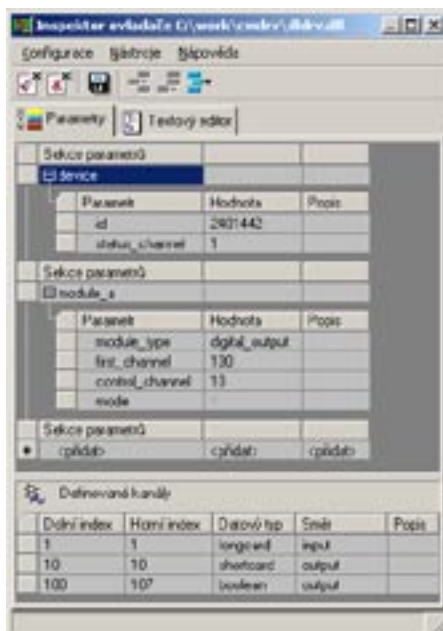
Univerzální OPC server podporuje všechna povinná rozhraní podle standardu OPC Data Access verze 2. Existují tedy čtyři základní způsoby výměny dat mezi serverem a klientem:

- Synchronní komunikace se zařízením je vždy blokující. To znamená, že požadavek klienta - čtení nebo zápis dat je dokončen, až jsou data přenesena z/do fyzického zařízení.
- Synchronní komunikace s vyrovnávací pamětí. Server poskytuje klientům hodnoty z vlastní vyrovnávací paměti. Nezávisle na komunikaci s klienty musí server zajistit periodické obnovování těchto hodnot.

- Asynchronní komunikace je neblokující. Klient pouze oznámí, které hodnoty chce číst nebo zapsat. Po dokončení komunikace server informuje klienta.

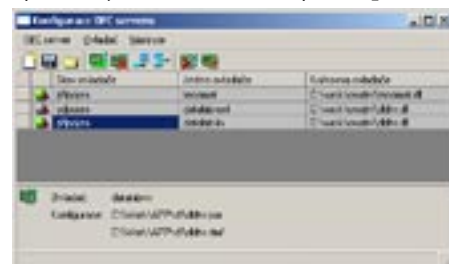
- Periodická komunikace. Server může sám periodicky číst data z připojeného zařízení a uchovávat je ve vyrovnávací paměti. Při změně některých hodnot může informovat klienty. Všechny typy komunikace je možné libovolně kombinovat. Jeden klient může část dat číst synchronně a část dat asynchronně. Je samozřejmě možné ještě před dokončením asynchronní komunikace zahájit další asynchronní nebo synchronní komunikaci.

Univerzální OPC server poskytuje rozhraní BrowseServerAdresSpace (v OPC specifikaci je označeno jako nepovinné). Toto rozhraní umožňuje klientům procházet celý adresní prostor serveru. Díky tomu je správné nastavení klientských aplikací velmi snadné. Adresní prostor univerzálního OPC serveru je vždy hierarchický (OPC specifikace umožňuje serverům definovat „ploché“ nebo hierarchický adresní prostor). Adresní prostor je rozdělen podle připojených zařízení. Datovými elementy (v OPC specifikaci „Items“) jsou kanály ovladačů připojených zařízení. Každý kanál je (stejně jako v celém systému Control Web) identifikován svým číslem a zpravidla odpovídá určité části adresního prostoru připojeného zařízení.



Součástí OPC serveru je velmi přehledné konfigurační rozhraní. Je zde možné

kdokoliv za běhu OPC serveru připojit nebo odpojit zařízení, případně změnit konfiguraci (například rozšířit rozsah komunikovaných elementů). Díky několika jednoduchým průvodcům a grafickému inspektoru ovladačů je konfigurace serveru velmi snadná. OPC server dále nabízí diagnostické rozhraní, které umožňuje online sledovat stav všech komunikovaných elementů, jejich hodnoty a požadavky na komunikaci ze strany klientů. Všechny chyby při komunikaci mezi serverem a klienty (například pokus o komunikaci neexistujícího elementu) jsou zaznamenávány. Díky tomu jsou diagnostika a sledování běhu celého systému jednoduché a vždy k dispozici.



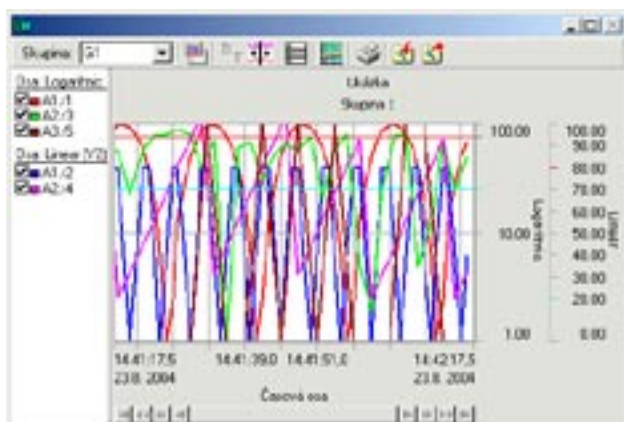
K OPC serveru můžeme připojit například jednotku DataLab IO s modulem 8 digitálních výstupů. Vytvoříme konfiguraci ovladače jednotek DataLab a kanály 101 až 108 budou odpovídat osmi digitálním výstupům. Ovladač připojíme k OPC serveru a nastavíme jeho jméno „Datalab_OUT“. Adresní prostor serveru je na obrázku. Jméno konkrétního datového elementu vytvoříme jednoduše ze jména ovladače a čísla kanálu. Například první digitální výstup je všem OPC klientům přístupný pod jménem „Datalab_OUT.101“.

Díky technologii DCOM (Distributed COM), je možné spustit univerzální OPC server na vzdáleném počítači. Protože DCOM je síťová technologie, je nutno jej správně konfigurovat z hlediska bezpečnosti. Uživatelé vytvářející OPC servery na vzdálených počítačích musí mít k těmto počítačům povolen přístup. Také ne všechny počítačové sítě umožňují provoz na IP portech, které DCOM využívá a otevření těchto portů může vyžadovat zásah správce sítě. To vše je nutno mít na zřeteli při nasazování OPC serverů přístupných přes DCOM.

Rozhraní ovladačů systému Control Web, které OPC server využívá, je otevřené a kompletně popsáno v dokumentaci. Díky tomu je tvorba OPC serveru pro libovolné zařízení velmi jednoduchá. Stačí vytvořit ovladač pro Control Web, ten navíc zcela jistě najde uplatnění při tvorbě vizualizačních aplikací.

Opravné balíčky pro systém **Control Web 5** občas nebývají jen prostými opravami nalezených chyb, ale mohou přinášet i velmi zajímavé nové komponenty a vlastnosti, které motivují uživatele, aby si update instaloval, i když zrovna na žádný problém ve své verzi nenaráží.

Rozšíření funkčnosti přístroje data_viewer



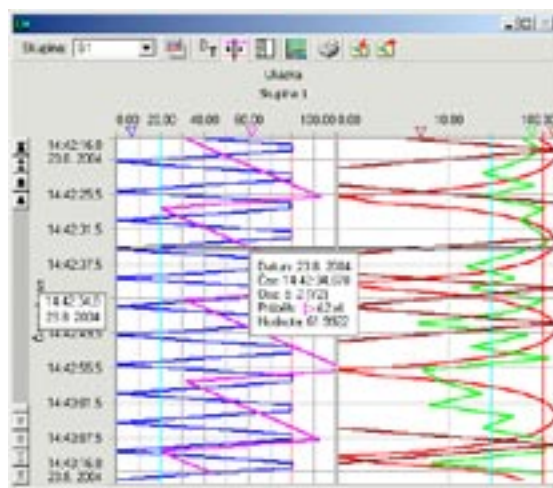
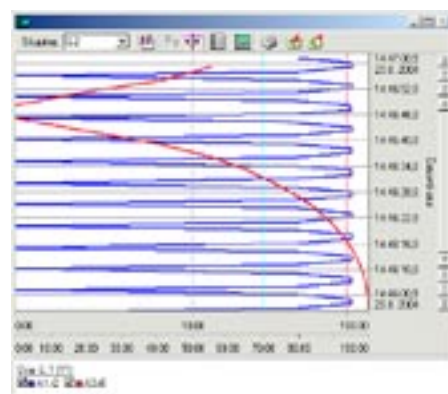
Přístroj data_viewer byl doplněn o řadu větších či menších rozšíření. Největší změnou je přidání svislých grafů. Spolu s tím byla přidána možnost definovat orientaci grafu. Svislé grafy tak lze kreslit jak shora dolů, tak zdola nahoru. U vodorovných grafů pak lze zadat orientaci zleva doprava nebo zprava doleva.

Legenda byla rozšířena o tlačítka pro zapínání a vypínání viditelnosti grafických průběhů, rolovací lišty a o automatické

uspořádání popisných textů do více sloupců (v závislosti na šířce legendy). Nově lze nastavit pozici legendy. Legenda tak může být umístěna na libovolné pozici vzhledem ke kreslicí ploše: vlevo, vpravo, nahoře i dole.

Podobně jako u legendy lze nastavovat i pozice os (y-ových i ča-

sově). V závislosti na orientaci grafu pak mohou být umístěny vlevo nebo vpravo, resp. nahoře nebo dole vzhledem ke kreslicí ploše. Časová osa byla dále rozšířena o možnost definovat tvar popisky (zda má obsahovat jenom čas, jenom datum či obojí). U mřížky byla doplněna možnost jejího překreslování přes průběhy.



Rozšíření funkčnosti přístroje httpd

Virtuální přístroj httpd byl velmi významně rozšířen. Přibyla řada nových parametrů i OCL procedur, které umožňují nasazení systému Control Web v roli podnikových WWW serverů. Zde uvedeme jen velmi stručný přehled novinek.

- Bylo změněno chování přístroje při obdržení HTTP požadavku POST. I po POST lze vrátit data klientovi obdobně jako po požadavku GET s daty v URL (podrobně je chování po POST popsáno v dokumentaci přístroje).
- Implementace protokolu HTTP byla rozšířena o možnost zasílat serveru soubory prostřednictvím požadavku POST dle RFC 1867 (Form-based File Upload in HTML).
- Přístroj httpd nyní kompletně spolupracuje s lokální vyrovnávací pamětí klientů a prostřednictvím hlaviček Last-Modified and If-Modified-Since redukuje síťový provoz (vrací 304 Not Modified, pokud má klient aktuální data).

Byla zpřísněna syntaktická kontrola při překladu přístroje. Pokud byly v přístroji zadány stejné identifikátory vyhledávané v HTML stránce, jejich náhrada způsobí nesprávnou funkci. Nově jsou detekovány i výskyty podřetězců a není nadále možno zadat dva různé řetězce, z nichž ale jeden je podřetězec druhého.

Nové parametry přístroje:

- Parametr ip_address umožňuje práci více serverů na počítači s více IP adresami.

Nové nativní procedury:

- PutFile umožní vložit do generovaného textu obsah souboru.
- RedirectToFile umožní přerušit dynamické generování dokumentu a odeslat soubor (např. obrázku) jako dokumentu.
- SetStatusCode umožní měnit stavový kód odpovědi HTTP. Umožní tak realizovat např. přesměrování apod.
- SetContentType umožní přímo měnit typ vrácených dat.
- SetLastModified umožňuje nastavovat

datum u dynamicky tvořených dokumentů.

- GetURLData zpřístupňuje původní řetězec dat předaný z formuláře prostřednictvím požadavku GET.
 - DateToString a StringToDate převádí datum a čas do a z textové podoby užívané v HTTP protokolu.
 - IPAddressToString a StringToIPAddress převádí číselnou podobu IP adresy (typu longcard) na textovou podobu (čtveřici čísel oddělených tečkami).
 - DecodeURL mění řetězec zakódovaný do podoby přenášené v URL požadavku do nezakódované podoby.
 - StripTags odstraní z řetězce HTML značky.
- Nové událostní procedury:
- OnFormData je vyvolána vždy když přístroj obdrží data z formuláře metodami POST i GET.
 - OnPostFile je vyvolána pokud klient odešle z formuláře soubor serveru (viz. RFC 1867).

Rozšíření funkčnosti 3D vykreslovacího systému

Ve 3D vykreslovacím systému byly některé existující mechanismy optimalizovány pro dosažení vyššího výkonu a některé nové vlastnosti byly přidány.

- Optimalizace načítání modelů ve formátu *.OBJ - soubory s 3D modely v tomto textovém formátu jsou často docela velké. Navíc je nutno je načítat dvouprůchodově. V tomto servisním balíčku bylo pomocí hluboké optimalizace paměťově mapovaných proudů dat dosaženo téměř desetinásobného zrychlení načítání těchto modelů. Nyní již v řadě případů zabere více času inicializační preprocessing dat než samotné načítání datových souborů. A toto předzpracování dat již asi znatelně urychlit nelze. U vizualizací s OBJ modely nyní zaznamenáme výrazně rychlejší start aplikačního programu.
- Načítání 3D modelů ve formátu *.3DS - systém byl rozšířen o načítání modelů v rozšířeném formátu 3DS. Načítání je rovněž optimalizováno na vysokou rychlost. Souřadný prostor modelu je patričnými transformacemi při načítání automaticky konvertován
- Bitmapové soubory *.CEL a *.PIC jako textury - možnosti načítání obrazových souborů jako textur jsou rozšířeny o tyto dva nové formáty.
- Možnost volby metody přenosu vektorových dat z CPU do GPU - pomocí nového parametru geometry_transfer_method můžeme definovat způsob přenosu geometrických dat do grafického procesoru.
- Nový příznak flares_disabled ve vlastnostech světelného zdroje - tento parametr umožňuje zakázat čočkové efekty pro zvolený světelný zdroj.
- Nová nativní procedura 3D přístrojů SetLightAbsPosition(LightOrder, X, Y, Z, W : real) - umožňuje nastavit pozici zdroje světla v absolutních souřadnicích prostoru scény (a nikoliv v relativních jednotkách, jejichž absolutní velikost je dána velikostí prostoru scény).
- Nové nativní procedury 3D přístrojů - vrací data daná obsahem sekce location - na vrácených datech se neuplatňují případné další transformace dané aktivitou evaluátorů nebo transformačních nativních procedur.
 - GetLocationTranslate(var X, Y, Z, : real) - vrací posun vůči počátku souřadného systému scény
 - GetLocationRotate(var Angle, AxisX, AxisY, AxisZ : real) - vrací natočení vůči orientaci souřadného systému scény
 - GetLocationScale(var X, Y, Z : real) - vrací poměr měřítka prostoru virtuálního přístroje vůči měřítku souřadného systému scény
- Pro 3D scénu (přístroj gl_scene) je možno definovat tzv. skybox - skybox je tvořen šesti texturami ve všech směrech uzavírajícími náš výhled do volného prostoru.



Modul analogových výstupů pro systém DataLab

Modul 12-bitových analogových výstupů s optickým oddělením doplňuje nabídku jednotek DataLab o možnost spojitého řízení ventilů, pohonů motorů atd.

Modul analogových výstupů má 8 napěťových výstupních kanálů v rozsahu 0 až 10V. D/A převodník má rozlišení 12 bitů, což odpovídá asi 2,5 mV krokům.


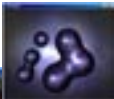




Čtyři kanály je možné pomocí propojek na desce přepnout do režimu proudových výstupů s rozsahy 0 až 20 mA. Vzhledem k vysoké přesnosti D/A převodníku je rozsah 4 až 20 mA řešen programově omezením rozsahu 0 až 20 mA.

Modul dodržuje limit maximálního příkonu pro napájení přímo ze sběrnice USB. Může tedy být použit i v malých jednotkách DataLab bez externího napájení.

Nové 3D virtuální přístroje

Pro zvýšení působivosti vizualizací (nebo také jen pro radost) bylo k systému přidáno několik nových dynamických 3D virtuálních přístrojů:

- gl_flag - přístroj jednoduše vytáhneme z palety a hned máme prapor vlající ve větru 
- gl_fluid_poster - proměnlivé pozadí s barevnými rozfázovanými texturami 
- gl_metaball - simulace kapalných objektů, je tvořena třírozměrnou sítí, v rámci které se potřebný počet sférických objektů může slévat, rozdělovat a měnit velikost 
- gl_water - simulace vodní hladiny, hladina se může vlnit, mohou se po ní šířit vlnky od dopadajících kapek deště a dokonce můžeme na hladině vytvářet vzruchy pomocí myši 

Možná zde bude užitečné opět připomenout, že celý 3D vykreslovací systém je dynamicky připojován až v okamžiku, kdy je to třeba. Pokud tedy komponenty z nějaké dynamické knihovny ve své aplikaci nepoužíváme, nespotřebávají tyto vůbec **žádné systémové prostředky**. Aplikační programy, které nepoužívají žádné 3D virtuální přístroje mohou být zcela bez problémů provozovány i na miniaturních vestavěných počítačích bez podpory 3D grafiky.



Kód	Produkt	Cena pro integrátory	Koncová cena	Výrobce
-----	---------	----------------------	--------------	---------

Control Web 5

CW5-DEV	Control Web 5 Vývojová verze	19 700 Kč	21 700 Kč	Moravské přístroje
CW5-UCW4	Control Web 5 Vývojová verze upgrade z Control Web 2000	12 900 Kč	14 200 Kč	Moravské přístroje
CW5-SRUN	Control Web 5 Runtime	5 900 Kč	6 500 Kč	Moravské přístroje
CW5-NRUN	Control Web 5 Runtime Network Edition pro provozování síťových distribuovaných aplikací	9 700 Kč	10 700 Kč	Moravské přístroje
CW5-TWCW	Control Web 5 Technologická záruka bude poskytnuta po předložení poukazu		500 Kč	Moravské přístroje
CW5-DEMO	Control Web 5 Demonstrační verze na CD-ROM lze zdarma stáhnout z http://www.mii.cz		250 Kč	Moravské přístroje

Ovladače dodávané spolu se systémem

CW5-NET	Ovladač pro komunikaci v síti přes protokol TCP/IP	zdarma	zdarma	Moravské přístroje
CW5-DLUSB	Ovladač pro komunikaci s USB moduly DataLab I/O vyžaduje Windows 2000 a vyšší	zdarma	zdarma	Moravské přístroje
CW5-ASCII	Univerzální ovladač pro textovou komunikaci přes sériové rozhraní RS-232	zdarma	zdarma	Moravské přístroje
CW5-DDECL	Univerzální DDE klient	zdarma	zdarma	Moravské přístroje
CW5-DLEIB	Ovladač rozhraní DataLab IF/EIB vyžaduje Windows 2000 a vyšší	zdarma	zdarma	Moravské přístroje

DataLab PC

DL-PC500	DataLab PC 500 256 MB PC133 SDRAM	10 700 Kč	11 750 Kč	Moravské přístroje
DL-PC600	DataLab PC 600 256 MB DDR266 SDRAM	11 900 Kč	13 100 Kč	Moravské přístroje
DL-PC610	DataLab PC 610 256 MB DDR266 SDRAM	13 400 Kč	14 750 Kč	Moravské přístroje

DataLab PC/IO

DL-IO500	DataLab PC/IO 500 256 MB PC133 SDRAM + CPU modul jednotky DataLab IO	12 970 Kč	14 250 Kč	Moravské přístroje
DL-IO600	DataLab PC/IO 600 256 MB DDR266 SDRAM + CPU modul jednotky DataLab IO	14 170 Kč	15 600 Kč	Moravské přístroje
DL-IO610	DataLab PC/IO 610 256 MB DDR266 SDRAM + CPU modul jednotky DataLab IO	15 670 Kč	17 250 Kč	Moravské přístroje

DataLab IO

DL-CPU	DataLab IO skříňka + CPU	2 970 Kč	3 250 Kč	Moravské přístroje
DL-CPUM	DataLab IO _μ skříňka + CPU	1 930 Kč	2 100 Kč	Moravské přístroje
DL-DI1	DataLab modul DI8 8 digitálních izolovaných vstupů	1 350 Kč	1 500 Kč	Moravské přístroje
DL-DI2	DataLab modul DI8SE 8 digitálních izolovaných vstupů, společná zem	1 350 Kč	1 500 Kč	Moravské přístroje
DL-DO2	DataLab modul DO8 8 digitálních izolovaných výstupů	1 350 Kč	1 500 Kč	Moravské přístroje
DL-DO1	DataLab modul RO8 8 releových výstupů, spínací	1 500 Kč	1 650 Kč	Moravské přístroje
DL-AI1	DataLab modul AI8 8 analogových vstupů, 16 bitů	2 690 Kč	2 950 Kč	Moravské přístroje
DL-AO1	DataLab modul AO8 8 analogových výstupů, 12 bitů	2 890 Kč	3 200 Kč	Moravské přístroje
DL-EIB	DataLab EIB EIB/USB	3 700 Kč	4 050 Kč	Moravské přístroje

Zde je uvedeno jen několik hlavních produktů. Další stovky položek naleznete na <http://www.mii.cz>. K dispozici je rovněž internetový obchod, který vám zjednoduší evidenci a umožní sledovat stav vašich objednávek (objednávka přijata, vyřízena, ...) a také si můžete zobrazit historii svých nákupů.

Moravské přístroje a.s.
Masarykova 1148
763 02 Zlín-Malenovice

<http://www.mii.cz>
<http://www.controlweb.cz>
<mailto:info@mii.cz>

tel./fax 577 107 171
tel. 603 498 498
tel. 603 228 976

