CONTROL WEB - OBJEKTOVÉ VÝVOJOVÉ PROSTŘEDÍ (NEJEN) PRO PRŮMYSLOVÉ APLIKACE

Jiří Kofránek

Oddělení biokybernetiky Ústavu patologické fyziologie 1. LF UK Praha, U nemocnice 5
128 53 Praha 2, e-mail:kofranek@cesnet.cz

Levné velíny z obyčejných PC

Díky masové výrobě je dnes "péčéčko" jednou z nejlevnějších periférií technologického zařízení. Proto se na trhu objevuje řada systémů, které slouží k rychlému a flexibilnímu vytvoření vizualizačních měřicích nebo řídicích aplikací (vzpomeňme např. na klasický systém LabView). Tímto způsobem je možné poměrně levně vytvořit "velín" z obyčejného PC, vybaveného příslušným adaptérem (I/O kartou), komunikující s technologickým zařízením.

Valná část těchto systémů pracuje na jednoduchém a léty ověřeném principu: vytvořený program pracuje cyklicky v určité smyčce, kde postupně obhospodařuje sadu záznamů, v nichž jsou uschovávána veškerá (vstupní či výstupní) technologická data. Při aktivaci těchto záznamů mohou být technologická data v číselné či grafické podobě vizualizována na obrazovce PC, či je naopak iniciován přenos řídících dat do připojené technologické aparatury. Systémy poskytují různé možnosti interaktivního vytváření grafické podoby obrazovek "velínu" z jednotlivých virtuálních přístrojů (měřících ručičkových či digitálních displejů, obrazovek pro průběžné zobrazování grafů, tlačítek, přepínačů apod.). Podstata "programování" pak spočívá v zařazování aktivace těchto přístrojů do nekonečné smyčky řídícího programu.

Problém je ale v tom, že obvykle není možné příliš ovlivnit rychlost, strukturu a mechanismus čtení a záznamů dat do vstupních a výstupních zařízení, což při menších objemech veličin příliš nevadí. Jakmile je však počet načítaných či odesílaných dat větší, dostává se systém do problémů – některé elementy jsou obsluhovány příliš často a na mnoho jiných se zase včas nedostává. Má-li systém pracovat v reálném čase, neznamená to, že stihne úplně všechno – program je snadno možno přetížit a pak se stane, že vnitřní smyčka nestihne včas doběhnout do potřebného začátku dalšího spuštění. Autor aplikace však musí mít časování systému zcela pod kontrolou a musí mít diagnostiku, která umožní vše monitorovat a aplikaci vyladit tak, aby se nedostávala do časového skluzu.

U cyklických systémů stačí definovat nějakou proměnnou a systém pak dělá to, co je v něm jeho autory pevně naprogramováno: tuto proměnnou pak systém může cyklicky číst nebo zapisovat do nějakého v/v zařízení.

Systém ***Control Web*** však zvolil složitější přístup: místo neovlivnitelného (pouze parametrizovatelného) vnitřního cyklu je možné vytvářet ***volně programovatelnou aplikaci***, kde dáme programátorovi do rukou možnosti určit, kdy se má ta či ona hodnota načítat či zapisovat do vstupně-výstupního zařízení, a to nejen pomocí nějakých pevných časových konstant, ale např. i jako reakce na některé události. Architektura systému je ***otevřená*** – systém se vytváří ze ***spolupracujících komponent, jejich sada není dopředu nijak omezena***. Navíc, jednotlivé komponenty (virtuální přístroje) mohou vzájemně spolupracovat přes TCP/IP protokol na intranetových a internetových sítích – v systému Control Web tedy můžeme vytvářet distribuované aplikace.

Control Web je tedy komponentové vývojové prostředí, které umožní flexibilně a rychle vytvářet distribuované vizualizační měřící nebo řídící aplikace. V následujících stránkách si stručně naznačíme jak.

Vizuální programování

Uživatelské rozhraní vývojového systému Control Web maximálně podporuje vizualizaci všech činností a možnosti tvorby programu grafickými prostředky. Uživatel nikdy není ponechán svému osudu nad prázdnou pracovní plochou. Tak jako v moderních programovacích prostředích i v Control Webu jsou k dispozici různí průvodci, které vytvoří kostru aplikace či nezkušenému programátorovi poradí s výběrem komponent, které je vhodné pro daný typ aplikace použít (obr. 1).

Program v Control Webu je možné vytvářet v grafickém prostředí, kde pomocí myši vybíráme jednotlivé komponenty a rozmisťujeme je na zobrazitelnou plochu či zařazením do stromů časování definujeme jejich aktivaci v programu z hlediska času (obr. 2). V inspektorech jednotlivých komponent interaktivně pomocí dialogů definujeme příslušné vlastnosti apod.

Zároveň ale můžeme celý systém překlopit z grafické do textové podoby a pokračovat v tvorbě programu v textovém režimu. Z textového režimu ale je vždy možné se kdykoli překlopit (po opravě všech případných syntaktických chyb) zpět do grafického režimu (obr. 3). Každou aplikaci je proto možně tvořit chvíli v textovém a chvíli v grafickém režimu, vždy podle potřeby. Obdobně – inspektory jednotlivých komponent (které lze vyvolat v grafickém režimu) nabízejí při definici jejích vlastností interaktivní dialog nebo možnost definovat vlastnosti komponenty v textovém editoru (obr. 4).

Propojení s vnějším světem

Aplikace napsaná v systému Control Web komunikuje s vnějším světem přes ovladač příslušného vstupně-výstupního zařízení prostřednictvím speciálních proměnných - tzv. "vstupních a výstupních kanálů", které se nadefinují pro daný ovladač (viz obr. 5). Systém poskytuje prostředky, kterými lze přesně řídit časování, kdy jsou potřebná data čtena či zapisována do jednotlivých kanálů. Pro ladění aplikace je k dispozici speciální inspektor, který "hlídá", zda a jak počítač "stíhá" čtení či zápis v požadovaných intervalech (obr. 7). Pokud počítač "nestíhá", dozvíme se to a musíme buď změnit aplikaci nebo jí provozovat na silnějším hardwaru, nebo ji rozložit jako distribuovanou aplikaci na více počítačů. Půvab systému Control Web totiž spočívá v tom, že jednotlivé komponenty různých programových modulů běžících na propojených počítačích spolu mohou jednoduše komunikovat. Tato komunikace je samozřejmě kryptovatelná, aby byla zajištěna maximální ochrana přenášených dat.

V Control Webu tak máme nástroj, který nám umožní bez velké námahy vytvářet distribuovaná řešení na standardním hardwaru, komunikujícím s vnějším světem přes stovky ba i tisíce vstupních a výstupních kanálů. Rozsáhlé, nyní reálně používané aplikace v Control Webu dnes využívají dokonce i více než deset tisíc kanálů (např. ve Škodovce).

Pohodlí, které pro vytváření uživatelského rozhraní nabízí Control Web v kombinaci se spolehlivostí a numerickou rychlostí vytvářeného programu však lze využít nejen pro průmyslové aplikace. My například využíváme Control Web pro tvorbu lékařského

simulátoru (Kofránek a spol. 1999, 2000)[1]. Protože vlastní model fyziologických regulací v lidském organismu, na němž je simulátor založen je poměrně náročný na numerickou rychlost výpočtu (39 nelineárních diferenciálních rovnic, 179 proměnných) a zároveň klade velké požadavky na bohatost a funkčnost uživatelského rozhraní (množství grafů i zobrazovaných hodnot v rozmanitých fyziologických schématech umístěných na mnoha panelech), při výběru prostředku pro tvorbu vlastního simulátoru jsme sáhli po softwarovém nástroji určeném pro tvorbu průmyslových aplikací. Právě v průmyslových aplikacích totiž nezřídka jde právě o kombinaci požadavků na názornost vizualizace i na dostatečnou numerickou rychlost.

Při tvorbě simulátoru jsme využili určitý trik: místo ovladače nějaké měřící/řídící karty jsme napsali ovladač "virtuální karty" v jehož jádře je vlastní simulační model. Systému Control Web se ale zdá, že komunikuje s nějakou skutečnou kartou – ve skutečnosti však výstupní kanály jsou vstupy do modelu a vstupní kanály jsou výstupy z modelu (obr. 6). Pak nám již nic nebránilo využít všech možností vytváření košatého uživatelského rozhraní s množstvím knoflíků, přepínačů, měřících přístrojů, grafů a dalších komponent, které systém Control Web nabízí.

Kostičky z komponentové skládačky – virtuální přístroje

Aplikační program v systému Control Web je sestaven z jednotlivých ***virtuálních přístrojů***. Těmito přístroji jsou různé typy zobrazovacích elementů (různé měřící přístroje, grafy, osciloskop, ikony apod.), řídící prvky (přepínače, tlačítka, knoflíky aj), archivační elementy, prvky pro obsluhu alarmů, ale i obecný kontejner pro zobrazení Active X komponenty) apod. Tyto přístroje je možné pomocí myši interaktivně rozmisťovat na jednotlivé panely do jednotlivých oken (viz obr. 2).

Každý virtuální přístroj má specifické ***editovatelné vlastnosti*** (které je ale možné prostřednictvím volání příslušných metod měnit i za běhu aplikace). Tak např. voláním specifické metody můžeme přístroj zneviditelnit, nebo naopak nezobrazený přístroj můžeme na obrazovce zviditelnit. V každém virtuálním přístroji můžeme definovat jeho (zvnějšku přístroje neviditelná) data (atributy) a příslušné procedury (metody) reagující na nejrůznější události – např. na aktivaci přístroje (viz obr. 4). Každý přístroj může ***posílat zprávy*** jiným přístrojům v aplikaci: může je např. aktivovat, nebo způsobit spuštění jejich specifických ***metod*** (procedura uvnitř jednoho přístroje může volat metodu – tj. lokální proceduru jiného přístroje).

Virtuální přístroj nemusí vždy být jen zobrazitelná komponenta. Krom viditelných virtuálních přístrojů se v aplikaci hojně využívají i neviditelné virtuální přístroje. Typickým "neviditelným" (lépe řečeno na obrazovce vytvářeného "velínu" nezobrazitelným) virtuálním přístrojem je tzv. přístroj "program". Program je v obecná komponenta, kde můžeme definovat její atributy – tj. lokální data (konstanty, automatické a statické proměnné) a její metody – lokální procedury komponenty. Krom procedury "OnActivate", vyvolávané vždy při aktivaci komponenty, můžeme nadefinovat řadu dalších procedur, které pak můžeme aktivovat odkudkoli z jiných komponent.

K dalším "neviditelným" komponentám např. patří různé typy integrátorů, regulátorů, ale třeba i ***"SQL-přístroj" pro komunikaci s databázemi*** přes ODBC rozhraní prostřednictvím SQL příkazů nebo ***internetový http server*** pro komunikaci přes webové prohlížeče.

[1] Vývoj simulátoru je podporován výzkumným záměrem MŠMT č. 111100008 (physiome.cz), grantem UK č. 217/98 a společností BAJT servis s.r.o.

Virtuální přístroje jsou sice samostatnými a na sobě nezávislými komponentami, ale mají-li dohromady tvořit soudržný a výkonný aplikační program, musejí být nějak propojitelné.

Základním propojením virtuálních přístrojů v aplikačním programu je struktura jejich časování a struktura viditelnosti. ***Struktura viditelnosti*** určuje, kde na obrazovce (v jakém panelu, v jakém okně) se bude dotyčný (zobrazitelný) přístroj nacházet a ***struktura časování*** stanovuje, kdy a za jakých podmínek bude přístroj aktivován. Tyto struktury jsou znázorněny v podobě hierarchických stromů v levé části grafického vývojového prostředí (viz obr. 3).

Každý přístroj má atribut "časovač" (timer), který určuje od kdy a s jakou periodou má být přístroj aktivován. Časovač ale může mít hodnotu "none" – což znamená, že aktivace přístroje je asynchronní – pak lze přístroj aktivovat pouze zvnějšku z jiných komponent nebo zásahem uživatele (např. stisknutí knoflíku pomocí myši). Obdobně periodicky časované přístroje můžeme soustředit do strukturovaných časovacích stromů, kde lze vyjádřit i složitý algoritmus časování (s využitím posloupností, podmínek i cyklů) – obr. 8.

Daty řízená aplikace

V Control Webu můžeme vytvářet dvojího druh aplikací - jednodušší, pouze ***datově řízené aplikace***, nebo (z hlediska programovací náročnosti) složitější ***aplikace reálného času***.

V prvém případě jsou aktivace jednotlivých virtuálních přístrojů řízeny změnou příslušných dat a asynchronními událostmi (stiskem tlačítek uživatelem). Jakmile se např. změní načítaná hodnota některého kanálu, okamžitě se aktivují přístroje, které s ní pracují, např. zobrazují její hodnotu nebo s ní počítají ve svých lokálních procedurách (metodách).

Datově řízená aplikace pracuje cyklicky – maximální možnou rychlostí jsou načítána data z technologie, podle změny načtených dat jsou pak postupně aktivovány jednotlivé přístroje, které s těmito daty pracují. Aktivované přístroje (nebo uživatelské zásahy) mohou aktivovat jiné přístroje, na něž je pak přeneseno řízení. V datově řízené aplikaci však přístroje mohou být aktivovány též implicitně: pokud některý přístroj svou činností změní některá globální data (tj. globální proměnné či kanály), s nimiž pracuje jiný přístroj, je tento přístroj automaticky aktivován.

Délku a četnost jednotlivých cyklů (čtení dat z technologie, aktivace přístrojů, zápis dat do technologie) jsou jádrem Contol Webu optimalizovány v širokých mezích a programátor je nemůže explicitně ovlivnit. Na druhé straně nemůže udělat chybu, která způsobí uváznutí systému – což se může snadno stát při nedbalém programování aplikací reálného času.

K čemu a kdy lze datově řízenou aplikaci využít:

- V případech, kdy chceme data z technologie archivovat a vizualizovat bez potřeby precizně řídit časování těchto dějů.
- Při vizualizaci pomalu se vyvíjejících dějů.
- Všude tam, kde přesné načasování měření dat je méně důležité než jejich přehledná grafická prezentace.
- Pokud potřebujeme rychle vytvořit aplikaci (např. pilotní funkční prototyp budoucí rozsáhlejší aplikace) – využíváme toho, že datově řízená aplikace realizuje mnoho věcí automaticky a návrh aplikace se tak usnadňuje.
- Tam, kde se převážně nejedná o aplikaci komunikující s technologií – v Control Webu je totiž možné pohodlně vytvářet i distribuované kancelářské a manažerské aplikace – s využitím prostředků bezpečné síťové komunikace, prostředků komunikace s databázemi i snadnosti tvorby uživatelského rozhraní pro různé typy uživatelů.

Jako hodinky…

Hlavní doménou uplatnění Control Webu jsou ovšem aplikace reálného času. Control Web zde poskytuje velké možnosti a prostředky pro optimální vyladění vytvářené aplikace vzhledem k výkonnosti hardwaru, na němž se aplikace provozuje. Na rozdíl od datově řízené aplikace je však náročnější na programátorskou práci.

Obecné schéma činnosti systému Control Web v aplikacích reálného času znázorňuje obr. 9.

Pro optimální řízení časového průběhu komunikace s technologií a časování jednotlivých komponent v Control Webu je vyhražen speciální prováděcí tok[2] - tzv. časovací prováděcí tok (viz obr. 9), který má nastavenou vyšší prioritu, než ostatní prováděcí toky ostatních aplikací systému Windows. Operační systém zaručuje, že ze všech toků, které jsou v daném okamžiku schopny běžet (tedy nečekají a nespí), poběží vždy tok s nejvyšší prioritou (a pokud je toků se stejně vysokou prioritou více, bude je operační systém střídat). Časovací prováděcí tok proto díky své prioritě poběží přednostně, právě tehdy, kdy bude potřeba.

Časovací prováděcí tok (časovací thread):

- zajišťuje periodické časování přístrojů, shromažďuje a rozděluje všechny asynchronní zprávy od přístrojů, ovladačů, vstupů od uživatele (a v datově řízené aplikaci se stará i o aktivaci komponent řízených změnami dat),
- přesně odměřuje délku komunikačních prodlev,
- neustále sleduje délku běhu jednotlivých přístrojů a ovladačů,
- v případné časové tísni zajišťuje popohnání aplikace vpřed.

Stručně a zjednodušeně řečeno, časovací prováděcí tok se v průběhu aplikace stará o přípravu jednotlivých elementárních běhů - ***časových kroků jádra systému Control Web***. Během těchto kroků (viz obr. 9) se vždy nejprve uskuteční vstupní komunikace s technologií – postupně se načítají hodnoty z technologie do vstupních kanálů a provede se postupná aktivace přístrojů. Každému kanálu může být (i dynamicky, prostřednictvím metody některého z přístrojů) nastaven maximální časový interval, po který se čeká na úspěšné načtení příslušných dat z přístroje. Nedojde-li během tohoto intervalu k úspěšnému načtení dat, pak se pokračuje se čtením dat z dalšího kanálu a přístroje v daném časovém kroku počítají se starou hodnotou kanálu. Nicméně pokus o čtení pokračuje na pozadí a v případě úspěšnosti Control Web registruje novou hodnotu, která se na konci probíhajícího časového kroku, nebo na začátku následujícího časového kroku zapíše do příslušného kanálu.

Po ukončení čtení časové jádro přistupuje k postupné aktivaci virtuálních přístrojů. Fronta přístrojů, čekajících na aktivaci může být v různých časových krocích různá (závisí to zejména na vlastním algoritmu periodického nebo asynchronního časování jednotlivých přístrojů, který má plně v rukou tvůrce aplikace). Každý přístroj může v průběhu své aktivace (a provádění svých metod) vyžadovat aktivaci jiného přístroje (resp. zasláním příslušné zprávy požadovat provedení některé jeho specifické metody). Tyto požadavky registruje časovací prováděcí tok, který naplánuje asynchronní aktivaci přístroje. Časový prováděcí tok rovněž registruje (a následně naplánuje do dalšího časového kroku) i další požadavky na čtení dat z technologie, které mohou přicházet z běžících metod (tj. lokálních procedur přístrojů). V Control Webu přitom máme programové prostředky, které po požadavku na čtení umožní proceduru přerušit a umožnit pokračování časového kroku jádra aktivací dalších přístrojů, a

[2] "Prováděcí tok" je nyní stále častěji používaný český název pro thread.

v dalším časovém kroku (kdy už byla požadovaná hodnota změřena) v provádění přerušené procedury pokračovat.

Délky časových kroků jádra mohou být různé – rozdíly mohou dosahovat až několika řádů. Závisí to na délce fronty aktivovaných přístrojů a především na výsledcích komunikace s technologií. Aplikace reálného času se přitom může dostat do skluzu (viz obr. 9). Časovací prováděcí tok se pak snaží skluz vyrovnat zkrácením intervalů, ve kterých krok jádra spí (využitím rezerv časových kroků).

Správně navržená aplikace by se ale do skluzu dostávat neměla. Control Web poskytuje řadu monitorovacích prostředků i programových nástrojů, které tvůrci aplikace poslouží k navržení takového uspořádání algoritmu, aby k častým skluzům nedocházelo a vytvořená technologická aplikace šlapala bezchybně a bez problémů jako hodinky.

Distribuované modulární aplikace

Program v Control Webu může být tvořen více moduly. Systém Control Web považuje za modul každý jeden aplikační soubor, takže je možné říci, že nejjednodušší aplikace je jednomodulární — taková aplikace se celá nachází v jednom jediném zdrojovém souboru. Výhodné je však rozdělit úlohu do více modulů – jednak z hlediska přehlednosti a v neposlední řadě i z hlediska možnosti týmové práce více tvůrců na jednom projektu. Výhodné je rozdělit aplikaci na rychlé moduly, které pracují s rychlými virtuálními přístroji (např. kreslení grafů) a na pomalé moduly, které pracují s pomaleji časovanými přístroji (např. s různými archivačními přístroji).

Při spuštění vícemodulární aplikace má každý modul svůj vlastní časovací prováděcí tok, který se stará o přípravu a provádění časových kroků jádra, které komunikují s technologií a obsluhují virtuální přístroje modulu. Každá komunikace s technologií je "poháněna" požadavky na čtení nebo zápis hodnot, které přicházejí z přístrojů – časovací prováděcí tok příslušného modulu pak tyto požadavky na začátku (čtení) i na konci (zápis) každého časového kroku jádra vyřizuje. U vícemodulární aplikace pak tyto požadavky mohou přicházet z přístrojů jiných modulů – a naopak: přístroj jednoho modulu může požadovat na jiném modulu poskytnutí hodnoty (nebo zapsání nové hodnoty) některého komunikačního kanálu a tím rozběhnutí příslušné komunikace s technologií (obr. 11). A nejenom to, krom hodnot kanálů či proměnných mohou spolupracující moduly navzájem posílat zprávy svým komponentám (tj. volat příslušné procedury virtuálních přístrojů).

Moduly nemusí běžet na jednom počítači – aplikace v Control Webu může být distribuovaná – jednotlivé moduly mohou běžet na mnoha počítačích propojených přes TCP/IP protokol. Distribuovaná aplikace v systému Control Web pak běží jako jeden celek, jako jedna velká logická aplikace. Právě distribuovanými aplikacemi je možno rozložit výkon na několik počítačů tak, aby se stíhalo komunikovat s rozsáhlejší technologií.

Moduly se spojují dovážením. Pokud např. modul A doveze modul B, pak modul A získá možnost používat prostředky modulu B. Moduly je možné dovážet i vzájemně — například dva moduly mohou dovážet jeden druhý a druhý zase první (zpětně). Pro dovozy modulů nejsou stanovena žádná omezení, takže vzájemné spojení modulů může vytvořit libovolnou strukturu (obecný graf).

Distribuovaná modulární aplikace pracuje jako jeden celek, lhostejno kde se jednotlivé moduly nacházejí. Při rozběhu se automaticky spustí všechny moduly na všech počítačích (tuto možnost zajišťuje trvale běžící proces - Control Web Daemon), a při ukončení distribuované aplikace se zase všechny běžící moduly na všech počítačích ukončí. Má to však jeden háček – pokud dojde k chybě v jednom z modulů propojených dovozem (např.

v důsledku výpadku příslušného počítače) – zastaví se celá distribuovaná aplikace i na ostatních počítačích.

Aby bylo možno předejít těmto problémům, je možné vytvářet propojení modulů nejen dovozem (zpřístupňujícím proměnné, komunikační kanály a přístroje), ale i tzv. připojením. Zastavení modulu, který se připojil k běžícímu modulu pouze ukončí spojení se vzdáleným počítačem a nezpůsobí ukončení běhu připojovaného modulu. V připojeném modulu je možno dosáhnout na všechny hodnoty globálních proměnných a hodnoty kanálů – a vyvolat tak v něm i požadavek na komunikaci s technologií, není však možné mu bezprostředně posílat zprávy jeho přístrojům (a vyvolávat jeho lokální procedury). Pokud však přesto chceme, aby přístroj v připojeném modulu mohl reagovat na nějakou zprávu z okolí – nic nám nebrání si v tomto přístroji naprogramovat reakci na protokol zpráv: přístroj bude sledovat nějakou sdílenou proměnnou a v závislosti na její hodnotě pak bude reagovat voláním některé své lokální procedury (metody).

Pomocí připojených modulů je možné např. vytvořit i tzv. "horkou zálohu" distribuované technologické aplikace. K záložní aplikaci připojíme vzdálený modul běžící aplikace a sledujeme hodnoty klíčových proměnných. Pokud vzdálená aplikace selže (výpadkem proudu aj.), záložní aplikace ihned přebere řízení. Jinou aplikací využívající připojování vzdálených modulů je systém pro dálkové lékařské vzdělávání, kterým se zabývá autor tohoto příspěvku (obr. 12).

Při vytváření distribuovaných aplikací jsou velmi důležitá i hlediska bezpečnosti. Control Web nabízí bezpečné a kryptované propojení přes intranetové či internetové TCP/IP sítě. Control Web umožňuje vytvořit pružný systém přístupových práv uživatelů – přístupová práva můžeme definovat ke každému modulu a ke každému přístroji.

Aplikace v Control Webu může komunikovat se vzdáleným okolím i pomocí jednoho ze svých standardních přístrojů – ***webového http serveru*** – přes něj může zviditelňovat data ze svých ostatních přístrojů či přijímat vstupy od uživatele.

Control Web není systém určený jen pro vytváření technologických aplikací. Nejen pro aplikace reálného času je např. užitečné, že Control Web umožňuje komunikovat přes ODBC rozhraní s nejrůznějšími databázemi (např. přes speciální přístroj SQL). S využitím speciálního přístroje - ActiveX kontejneru můžeme do systému snadno zakomponovat jakékoli ActiveX komponenty (a přes ně využít další aplikace vytvořené pro WIN32) – autor tohoto příspěvku např. využívá ActiveX kontejneru pro implementaci multimediálních animací do výukových programů pro distanční vzdělávání, vytvářených v systému Control Web.

Distribuované aplikace, tak snadno vytvářené v prostředí Control Web, zdaleka nejsou jen doménou průmyslu. Myslím, že Control Web najde i uplatnění pro řadu distribuovaných manažerských a kancelářských aplikací, zvláště, když v těchto aplikacích občas potřebujeme vizualizovat komunikaci s nějakou technologií (příkladem jsou např. zdravotnické informační systémy).

Z Moravy až do Japonska

Systém Control Web (nyní ve verzi 2000) je výsledkem dlouholetého cílevědomého úsilí zlínského týmu programátorů, kteří v roce 1990 založili vývojářskou firmu ***Moravské přístroje a.s.***

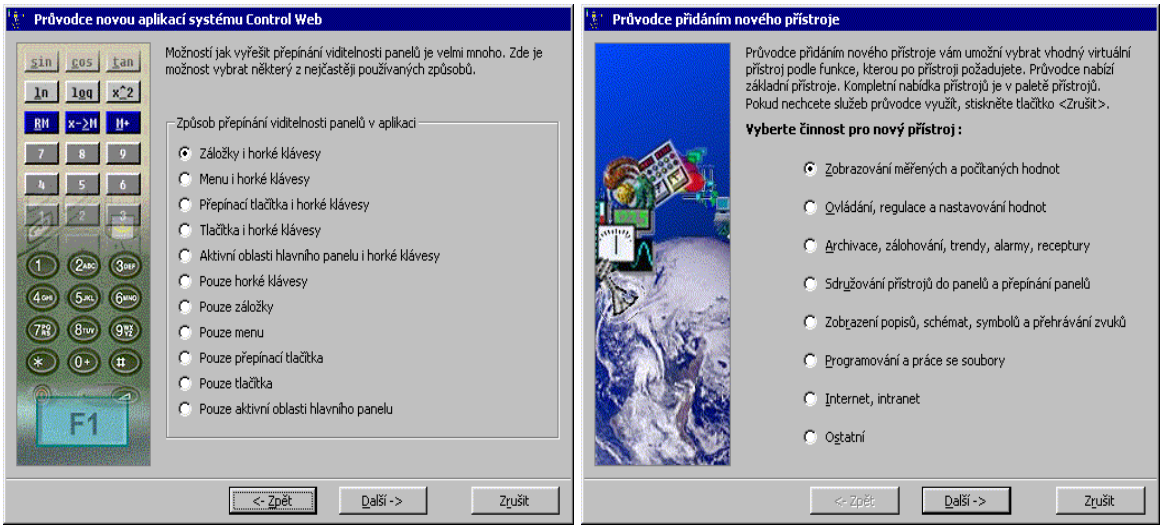
Před lety, v době šestnáctibitových Windows 3.1 vytvořili plně 16 bitový okenní systém InView, pracující v chráněném módu s vlastními paralelně pracujícími aplikacemi (textovým procesorem, tabulkovým kalkulátorem, kreslícím programem aj.), který měl s DOSem

společný pouze souborový systém. Od různých "recenzentů" v českých počítačových médiích tenkrát vesměs sklidili výsměch ("česká konkurence Microsoftu?!"). Program ale byl vedlejším produktem programového prostředí Control Panel s nástroji pro tvorbu aplikací pro řízení technologických procesů. Pro tyto oblasti nasazení jsou nutné systémy podporující chráněný mód procesoru, víceúlohové zpracování, výkonnou grafiku pro vizualizaci dějů a okenní systém pro ovládání. Tyto požadavky tehdejší Windows 3.11 nesplňovaly a tak si zlínská parta vývojářů naprogramovala vlastní operační systém. Portaci svého Control Panelu do prostředí Windows provedla až s nástupem jádra Win32 – do Window 95 a Windows NT.

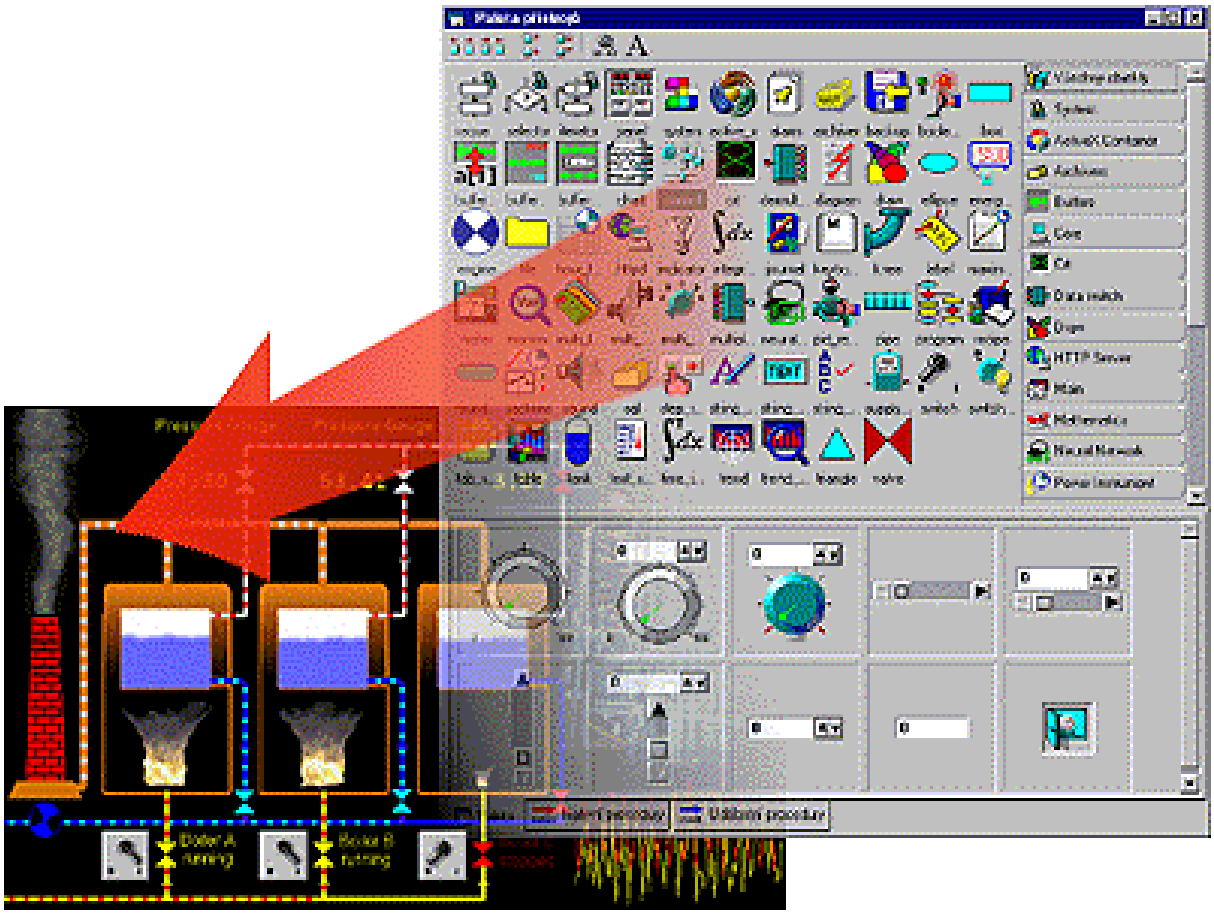
Současný produkt Control Web 2000 navazující na předchozí téměř desetiletý vývoj má nyní ve zdrojovém tvaru téměř milión řádek textu programu. Je distribuován v české, anglické, německé a nyní, díky zájmu distributorů z dálného východu, i v japonské verzi. V přípravě je i mutace do švédštiny. ***Je příkladem toho, že i v tuzemských podmínkách lze vytvářet softwarové produkty, které, přes všechny škarohlídy, si svojí kvalitou cestu na světový trh nakonec přece jen prorazí.***

Literatura

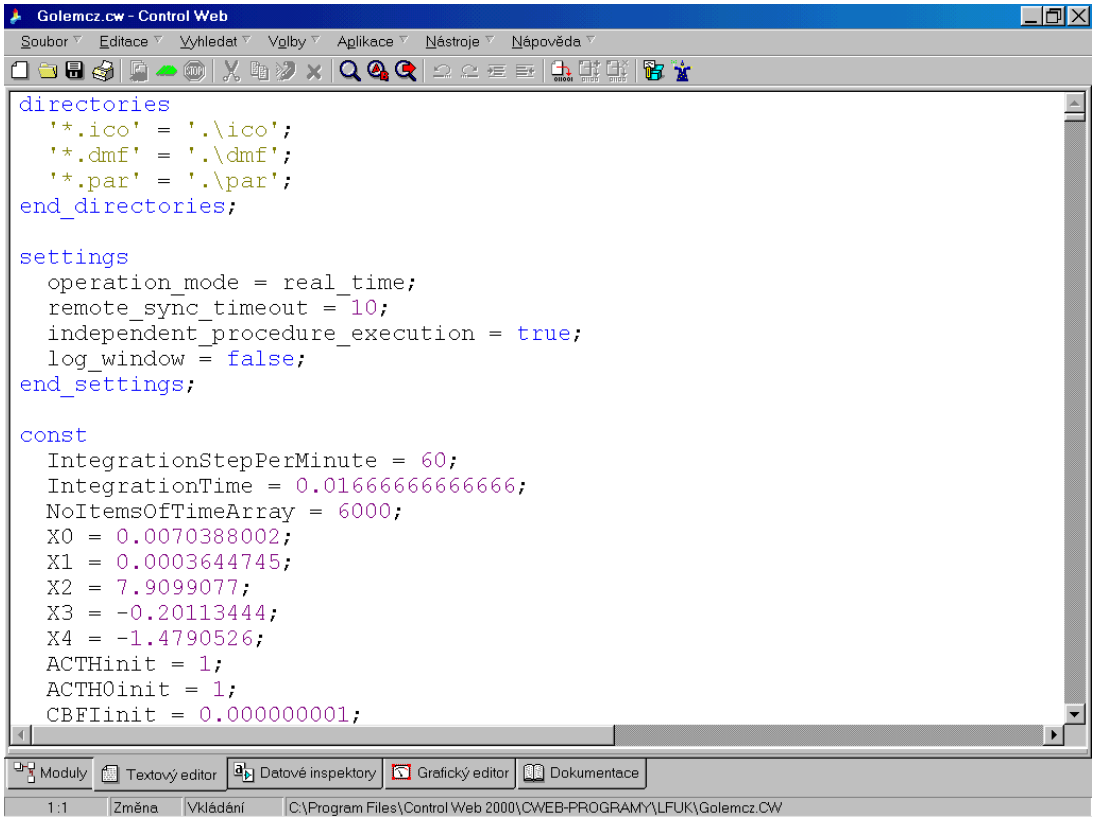
1. Jiří Kofránek, Tomáš Velan: Komponenty pro Golema. Využívání Simulinku a Control Webu při tvorbě simulátoru fyziologických funkcí. In Objekty' 1999, sborník 4. ročníku celostátní odborné konference (editor: V. Merunka, V. Sklenář). Česká zemědělská univerzita, Praha 1999, ISBN 80-213-0552-5, str. 189-204.
2. Jiří Kofránek, Tomáš Velan, Patrick Janicadis: Golem – Computer simulator of body fluids and acid-base disorders as an effitient teleeducation tool. In Modelling and Control Biomedical Systems 2000. IFAC Symposium. (editor: E. Carson, E, Salzseider), Pergamon, Elsevier Science, Oxforg 2000, str. 233-242.
3. Jiří Kofránek, Tomáš Velan, Patrick Janicadis: Šém pro Golema – zkušenosti s tvorbou a využitímvýukového simulátoru fyziologických funkcí.. In Medsoft, 2000 (editor: J. Cirýn), Tech-Market, Praha, 2000, str. 79-83.



Obr. 1 Control Web nabízí mnoho průvodců pro tvorbu aplikací, volbu jednotlivých komponent apod.

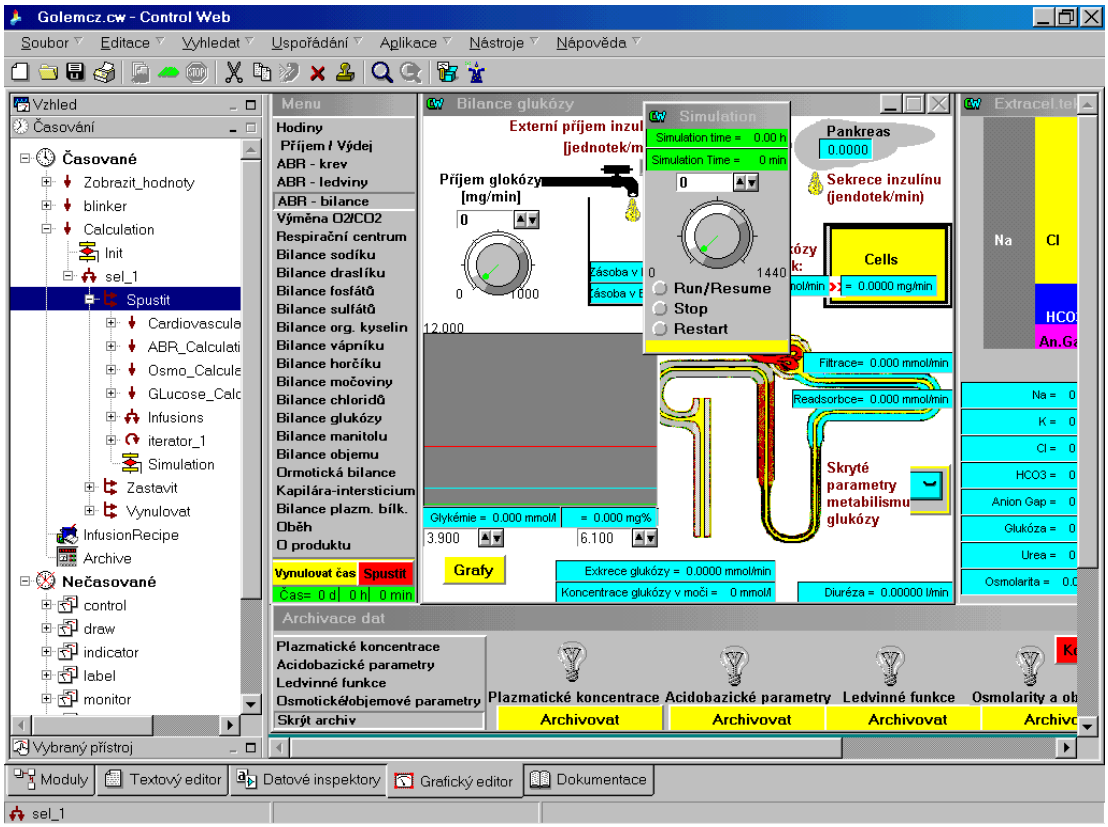


Obr. 2 Control Web plně podporuje vizuální programování. Aplikační program je sestavován v virtuálních přístrojů dostupných v "Paletě přístrojů", které jsou myší přetahovány do příslušné aplikace.



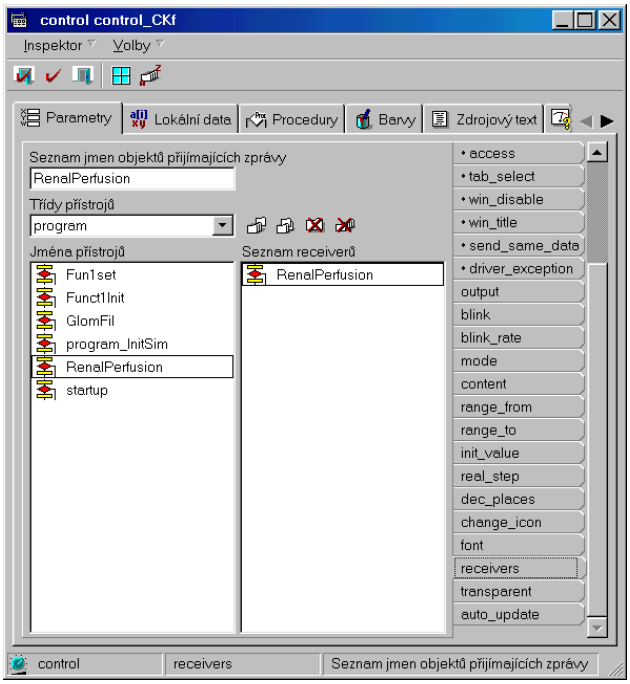
Textový režim

PŘEKLÁPĚNÍ

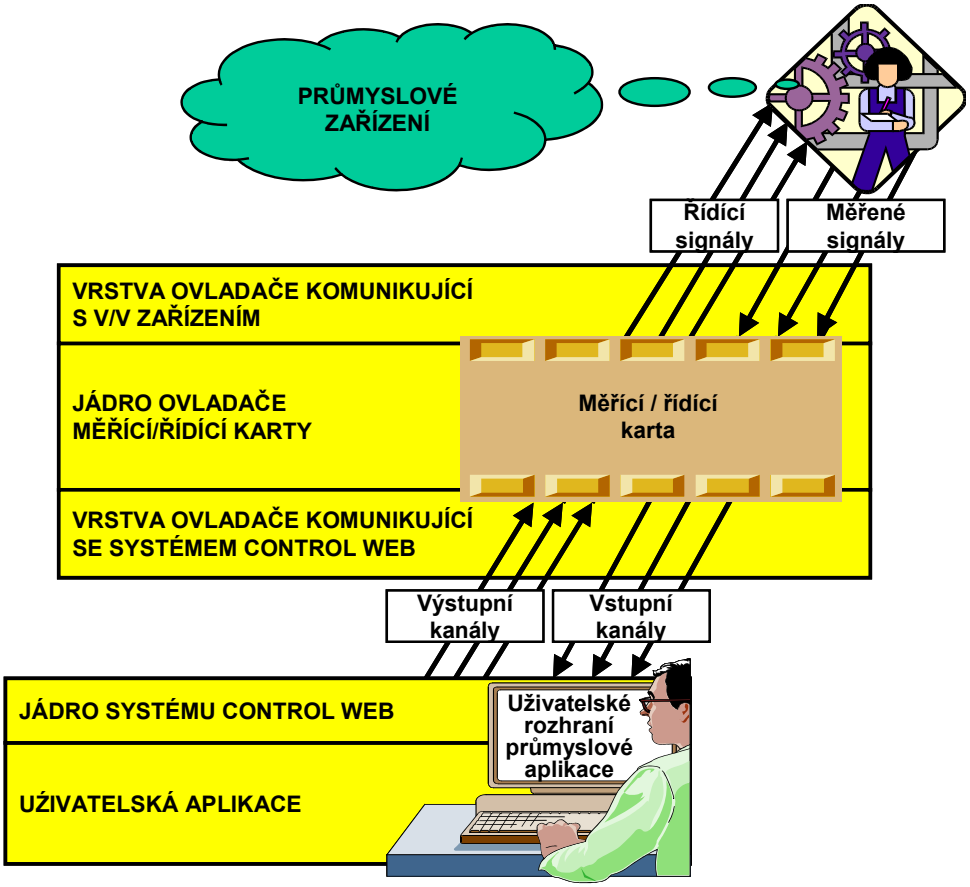


Grafický režim

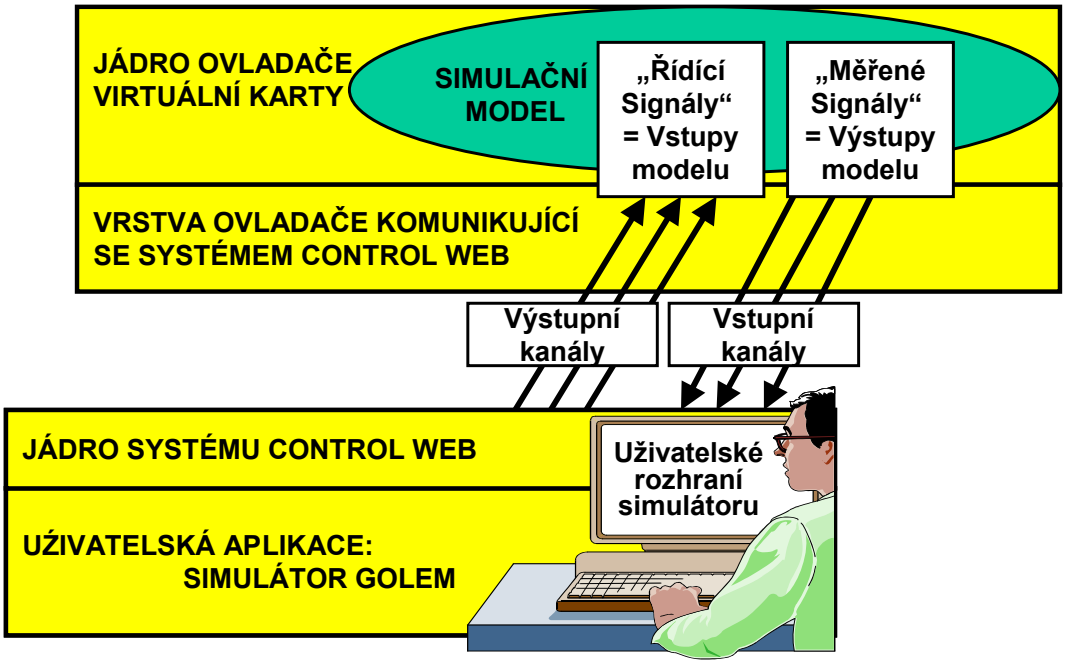
Obr. 3 Program může být vytvářen v grafickém nebo v textovém režimu. Mezi oběma režimy práce je možno jednoduše přepínat.



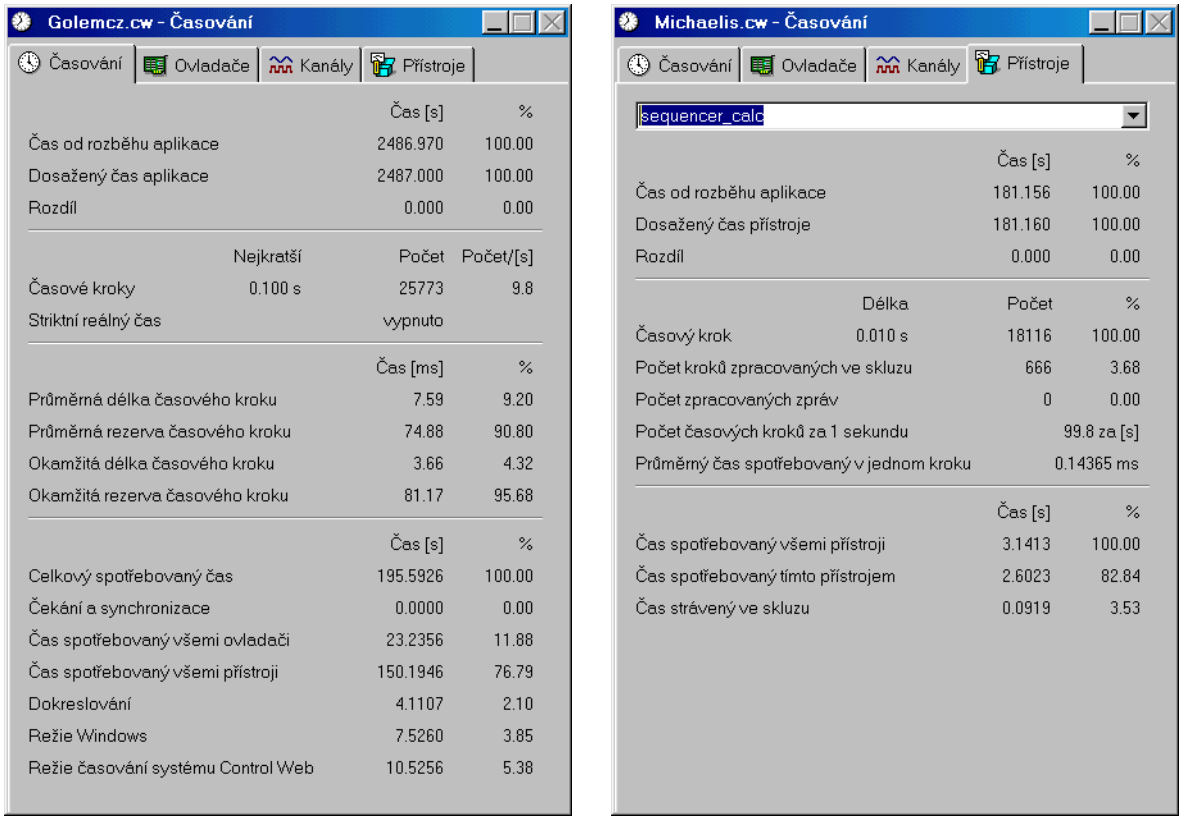
Obr. 4 V inspektorech jednotlivých komponent (virtuálních přístrojů) definujeme jejich specifické vlastnosti (např. periodicitu aktivace, určitý způsob zobrazení aj.). Můžeme definovat jejich lokální data (atributy) i procedury (metody), a definovat tak specifickou reakci komponenty na zprávy, které dostává (např. pomocí procedury OnActivate můžeme definovat chování komponenty při její aktivaci). Komponenta může zároveň vyslat zprávu jiným komponentám: aktivovat je, či voláním příslušné procedury (metody) měnit jejich vlastnosti apod.



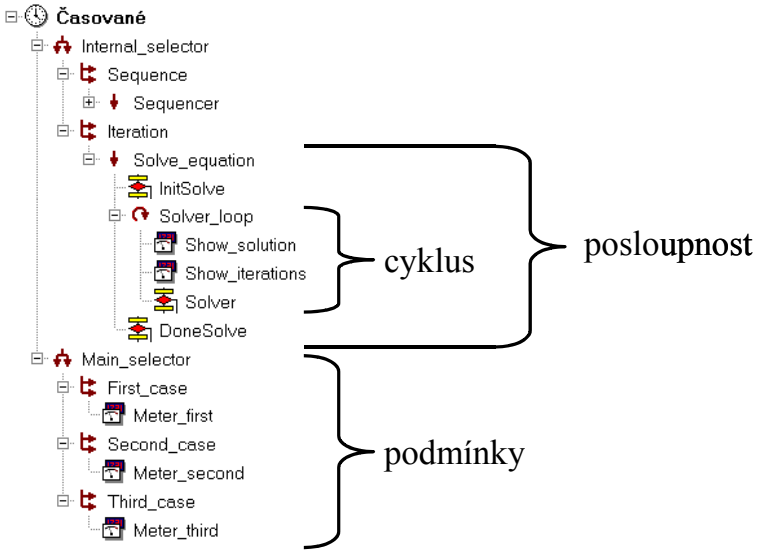
Obr. 5 Schéma propojení aplikace systému Control Web s vnějším světem



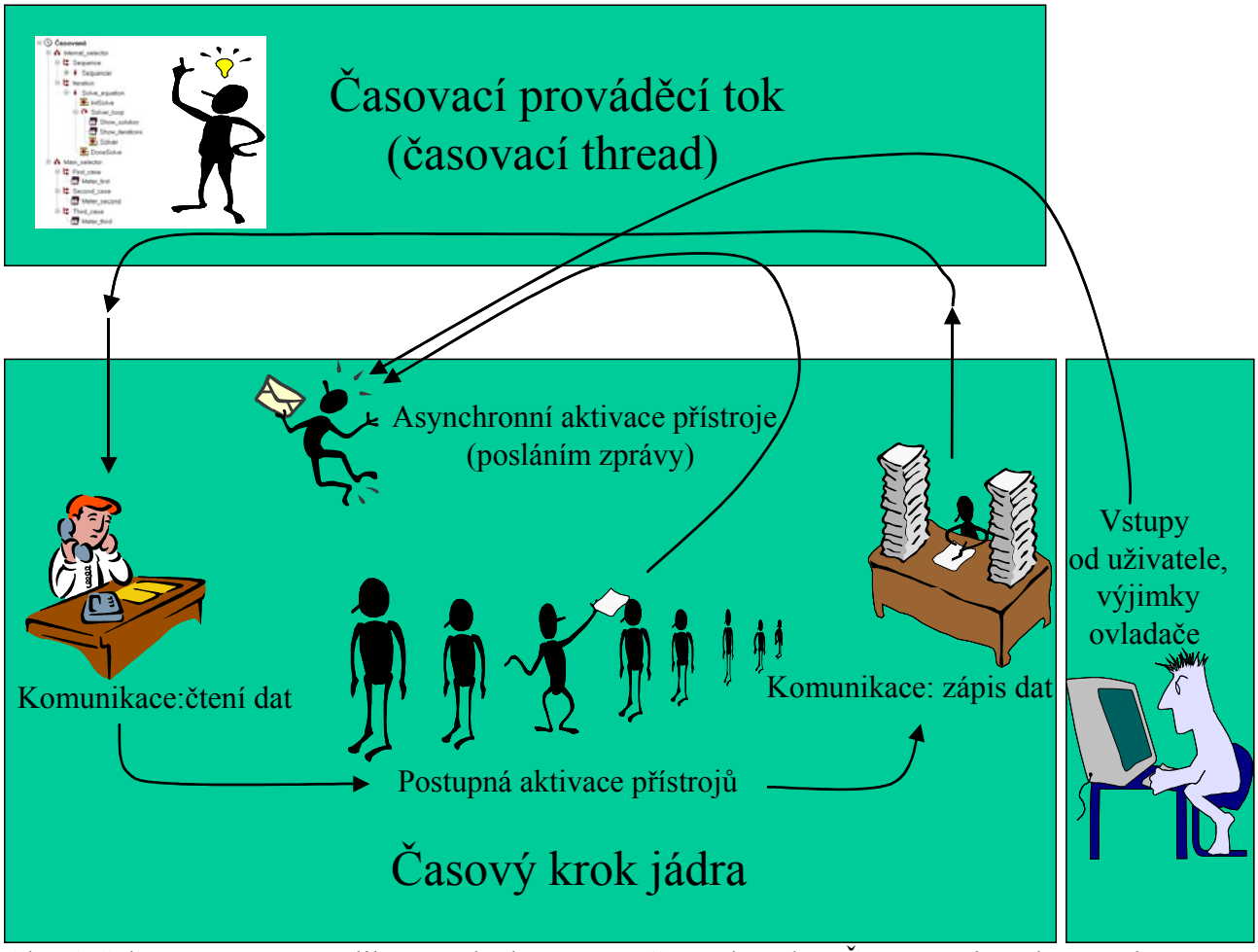
Obr. 6 Specifické využití systému Control Web autorem tohoto příspěvku: začlenění simulačního modelu do ovladače "virtuální karty" při tvorbě lékařského simulátoru.



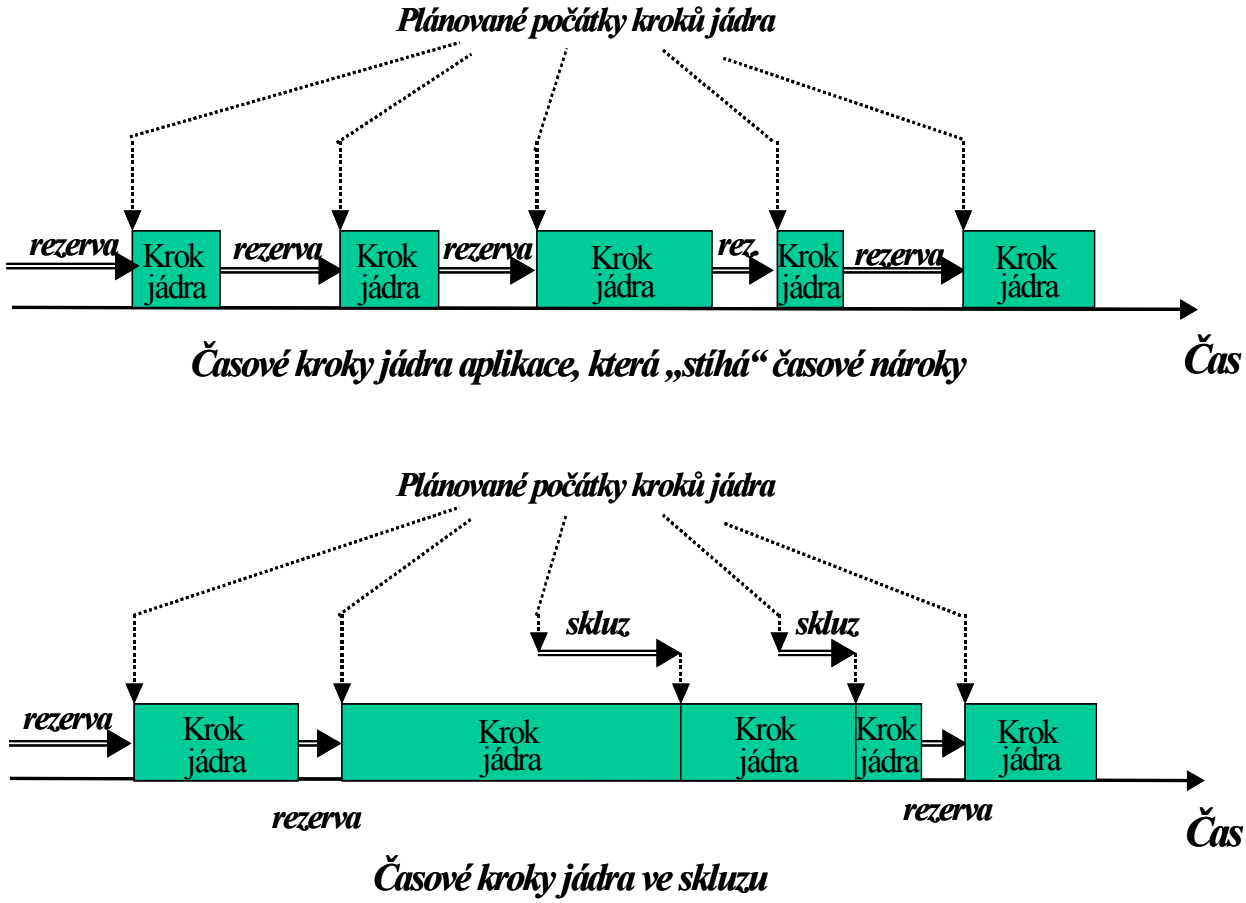
Obr. 7 Při ladění aplikací reálného času na PC je důležité mít přehled, zda a jak vytvořená úloha "stíhá" v definovaném čase komunikovat s technologií. Control Web nabízí prostředky pro podrobné sledování časové náročnosti úlohy až do úrovně jednotlivých virtuálních přístrojů.



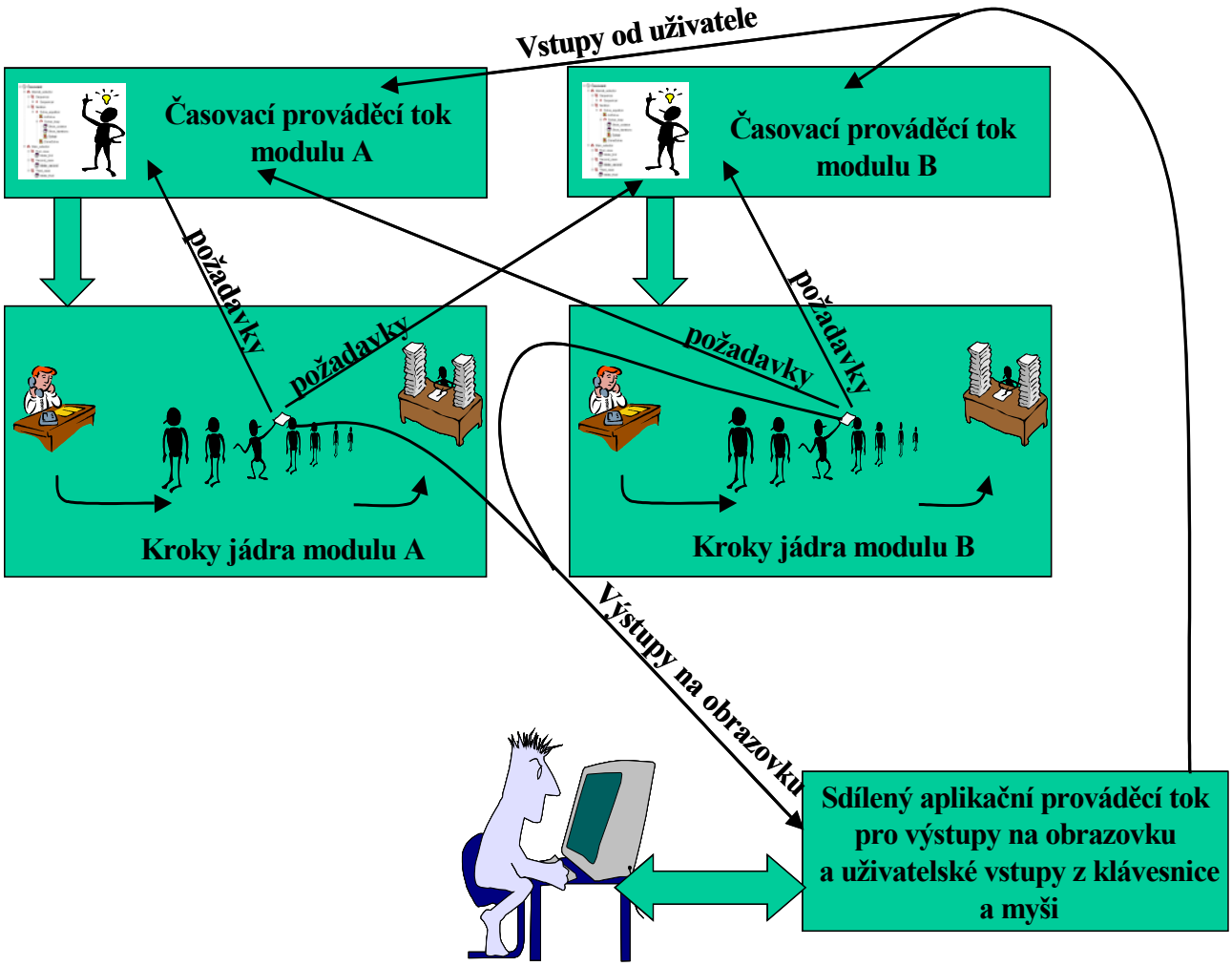
Obr. 8 Strom časování – přístroje s obdobným nastavením periodicity časování je možné soustředit do příslušných strukturovaných časovacích stromů. Pro vyjádření algoritmu časování lze využít posloupnost, cyklus se testovanou podmínkou na začátku a podmínkové selektory.



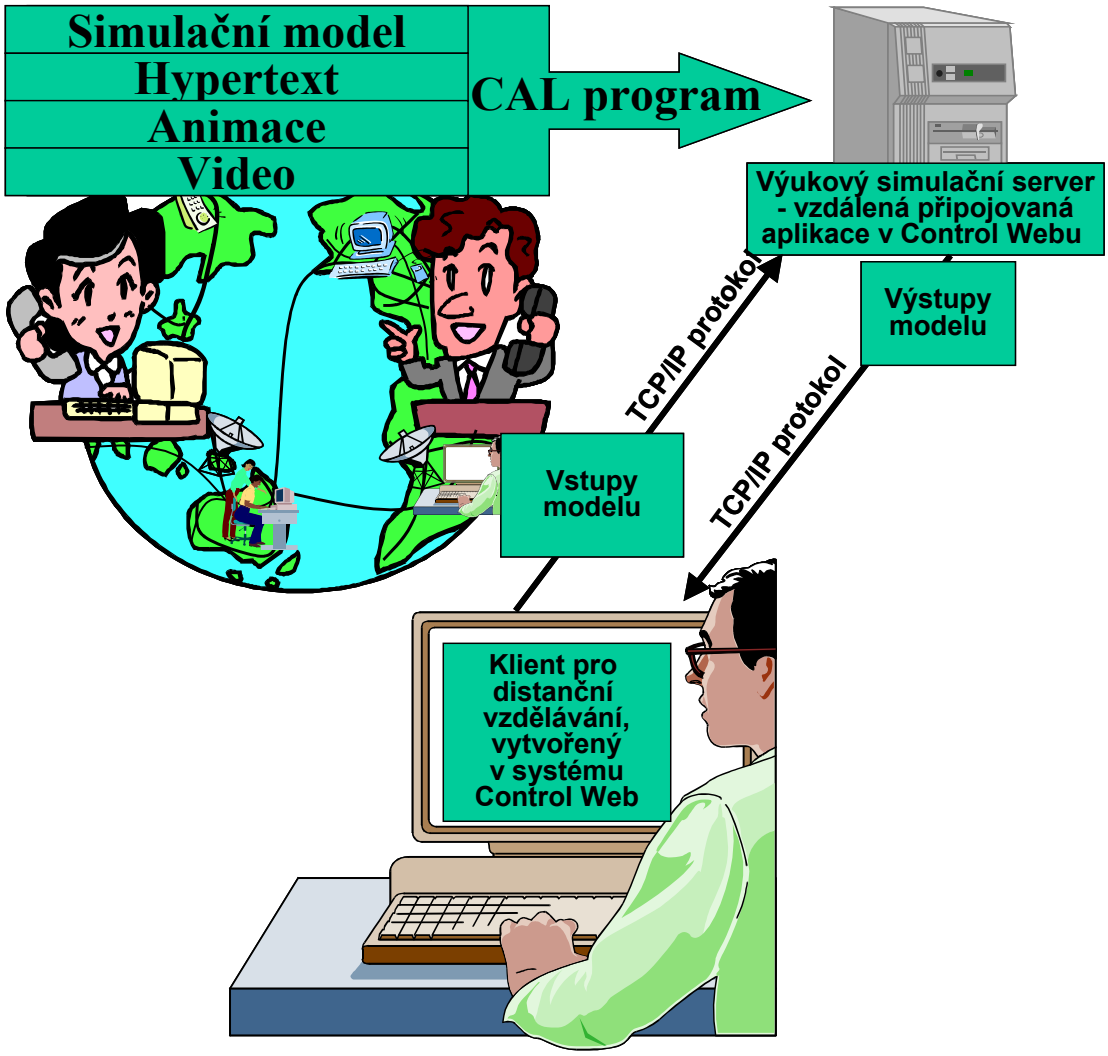
Obr. 9 Schéma časování aplikací reálného času v Control Webu. Časovací thread s vysokou prioritou nastavuje a řídí časové kroky jádra systému Control Web. Každý časový krok jádra začíná čtením dat z technologie, pokračuje postupnou aktivací přístrojů a končí zápisem dat do technologie. Přístroje mohou být aktivovány i asynchronně – jiným přístrojem, výjimkou z ovladače v/v karty či zásahem uživatele.



Obr. 10 Časové průběhy aplikace reálného času. Aplikaci se snažíme navrhnout tak, aby se nedostávala do časového skluzu. Control Web poskytuje prostředky pro monitoring skluzu aplikace (viz obr. 7).



Obr. 11 Vícemodulární aplikace: každý modul má svůj vlastní časový prováděcí tok, který "pohání" aktivitu jeho přístrojů. Požadavky na čtení či zápis dat je možno směrovat k jiným modulům. Aplikace může být i distribuovaná – moduly mohou být na různých počítačích. Všechny moduly na jednom počítači sdílejí společný prováděcí tok, který se stará o komunikaci s uživatelem.



Obr. 12 Využití distribuované aplikace vytvořené v systému Control Web pro distanční vzdělávání. Výukový CAL(computer aided learning) program, kombinující hypertext a multimedia se simulačními modely běží na serveru, ke kterému mohou být připojováni klientské programy pro distanční vzdělávání.