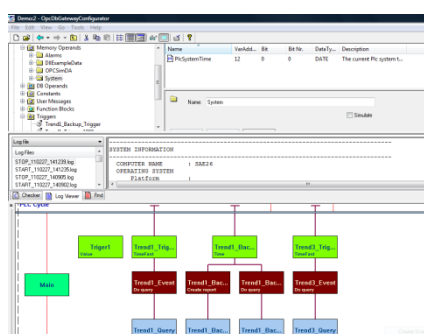


Integrácia aplikácií s OpcDbGateway

OpcDbGateway umožňuje integrovať zložité aj jednoduché softvérové aplikácie spracávajúce informácie z rôznych pripojených zariadení, databáz, súborov, webových služieb, softvérových aplikácií, skriptov a ukladať tieto informácie do databáz, súborov prípadne



Obrázok 1 Konfiguračná aplikácia

poskytovať spracované informácie externým zariadeniam cez rôzne komunikačné protokoly. Môže taktiež spúšťať rôzne externé programy s parametrami nakonfigurovanými pomocou konfiguračnej aplikácie OpcDbGateway a tiež uložené procedúry na databázach. Konfiguračná aplikácia OpcDbGateway (Obrázok 1) umožňuje minimalizovať čas implementácie aplikácie tým, že práčne programovanie umožňuje nahradiť konfigurovaním. Konfigurácia je potom uložená do konfiguračnej databázy, ktorá slúži pre riadenie výkonnej aplikácie OpcDbGateway.

Množstvo konfigurovateľných komponentov je možné ďalej rozširovať podľa individuálnych potrieb. Existuje možnosť naprogramovať si sadu vlastných modulov (rozširovacích DLL) pomocou programátorských jazykov ako napr. C++ alebo C# a, pri zachovaní definovaného programátorského rozhrania (API), sa tieto stávajú súčasťou konfigurovateľnej funkcionality OpcDbGateway. Takto je možné konfigurovateľnú funkcionality rozširovať neobmedzene. Rozšírenie funkcionality pomocou rôznych skriptovacích jazykov, či už proprietárnych alebo štandardných, je bežné aj v mnohých iných SCADA systémoch, avšak tieto je potrebné sa naučiť, a okrem toho nie vždy zabezpečujú (na rozdiel od modulov programovaných štandardnými programovacími jazykmi) dostatočnú výkonnosť.

OpcDbGateway pre komunikáciu s externými zariadeniami využíva prednostne komunikačné rozhranie podľa štandardu OPC. OPC servery, ktoré sa využívajú pre komunikáciu s externými zariadeniami pomocou proprietárnych protokolov zariadení a s OpcDbGateway komunikujú cez OPC rozhranie pomocou OPC klienta v OpcDbGateway, sú zvyčajne dodávané taktiež s vlastnými konfiguračnými aplikáciami. Pre odstránenie duplicitnej konfiguračnej práce, konfiguračná aplikácia OpcDbGateway umožňuje mapovať konfiguráciu adresného priestoru externého OPC servera do svojej konfigurácie.

Existujú tu tiež ďalšie nástroje integrátorskej práce – prostriedky na kontrolu konfigurácie, jej grafické zobrazenie, sledovanie behu integrovanej aplikácie pomocou logovania s nastaviteľnou úrovňou sledovaných detailov.

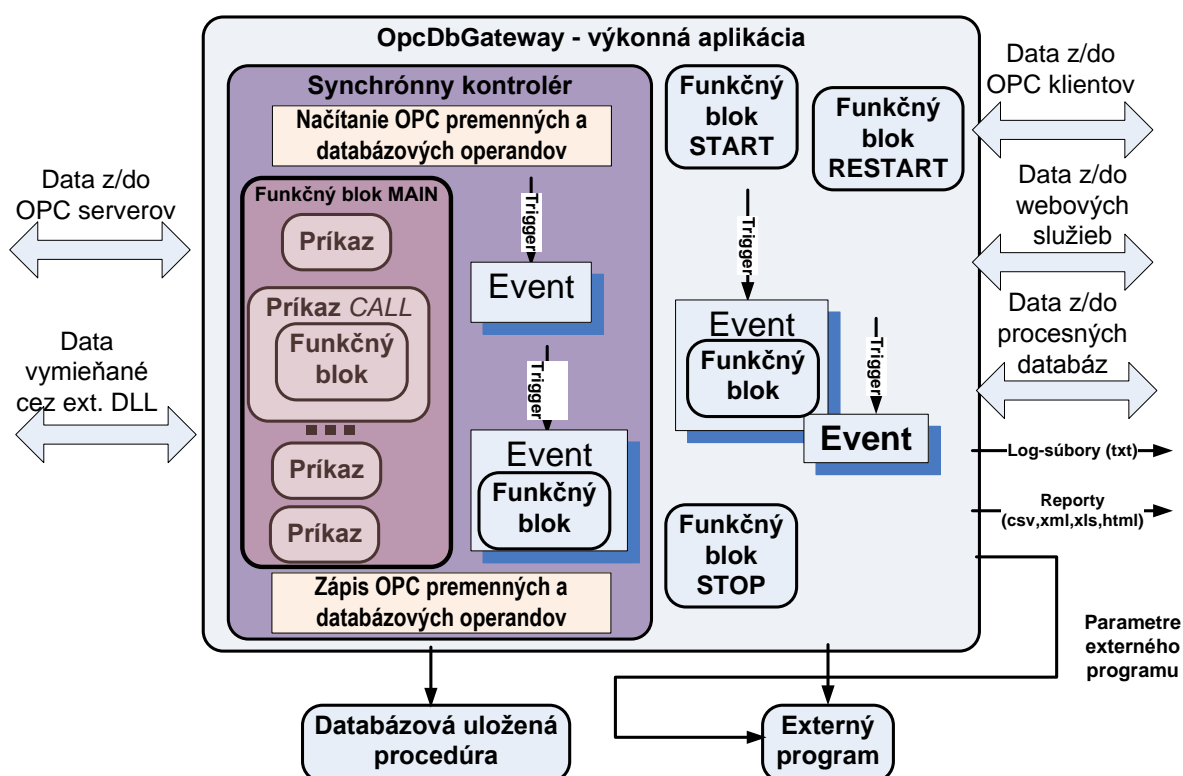
Opakujúce sa úlohy sú uľahčené s využitím softvérových pomocníkov (wizardov).

Synchrónne a asynchrónne výpočty

V rámci jednej aplikácie implementovanej pomocou OpcDbGateway môže byť zabezpečené **synchrónne** aj **asynchrónne spracovávanie úloh**. Každé z nich je vykonávané v osobitnom programovom vlákne (Obrázok 4). Synchrónne výpočty sú vykonávané v synchrónnom kontroléri (Obrázok 2).

Cieľom synchrónneho spracovania je vytvoriť prostredie pre vykonávanie **kvázi paralelných algoritmov** podobne ako je tomu napr. pri PLC používaných pre riadenie technologických procesov. PLC boli pôvodne vytvorené ako náhrada rôznych reléových riadiacich systémov realizovaných hardvérovo, u ktorých existoval skutočný paralelizmus vo funkcionalite jednotlivých funkčných častí. Ak sa má podobná funkcionalita zabezpečiť v jednej procesorovej jednotke, kde v konečnom dôsledku existuje iba sériové spracovanie, potom je potrebné zabezpečiť, aby sa **konzistentná zostava vstupných dát** počas spracovania transformovala počas **definovanej výpočtovej periódy** na **konzistentnú zostavu výstupných dát**. Ak je výpočtová perióda dostatočne krátka – zanedbateľná vo vzťahu k časovým konštantám riadenej sústavy, potom sa pre túto sústavu môže spracovanie údajov javiť ako paralelné. Synchrónne spracovanie prebieha teda cyklicky v troch fázach:

- Načítanie vstupných dát
- Výpočty
- Aktualizácia výstupných dát.



Obrázok 2 Integrátorský model OpcDbGateway

Pri výpočte sa využívajú samozrejme rôzne pomocné vnútorné dáta, ich transformácie však, z pohľadu riadenej sústavy (externých pripojených zariadení), nie sú relevantné. V synchrón-

nom režime OpcDbGateway načítava vstupné dáta z rôznych zariadení alebo dátových zdrojov, spracováva ich počas definovanej periódy a ukladá výstupné dáta do procesných databáz alebo posiela na rôzne zariadenia. Tento spôsob spracovania je, okrem náhrady reléových logík a integrovaných logických obvodov, vhodný tiež [pre riadenie spojitých sústav a procesov](#), včítane mnohoparametrových, [s časovými konštantami podstatne väčšími ako je perióda výpočtu](#). Dá sa využiť tiež napr. pri simulácii dynamických sústav alebo na monitoring a zber intervalových údajov z meračov spotreby energií.

[Špecifikom synchronného spracovania](#) je tiež synchronne načítanie [úplnej zostavy vstupov](#) z externých zariadení alebo databáz na počiatku cyklu a nastavenie [úplnej zostavy výstupov](#) na konci cyklu. Integrátor pri využití synchronného spracovania má do určitej miery zjednodušenú tvorbu konfigurácie v tom, že načítanie a aktualizáciu veľkého množstva premenných externých zariadení nemusí konfigurovať. Mal by ju však poznať, aby posúdil jej vhodnosť pre danú aplikáciu.

[Asynchronné spracovanie](#) je vhodné tam, kde periodicita transformácie vstupných dát na výstupné nie je podstatná. Napr., v čase vzniku mimoriadnej situácie v riadenom systéme potrebujeme zaznamenať stav systému do log-súboru, alebo keď je potrebné riadiť postup spracovania údajov („workflow“) v čase alebo na základe splnenia definovaných podmienok. V OpcDbGateway platí že, na rozdiel od synchronnej funkcionality, kde sa počítalo najmä so spracovávaním veľkého objemu dát naraz, je možné pristupovať k rôznym dátam z externých zariadení selektívnejšie. Z pohľadu integrátora to však môže byť o niečo prácnejšie.

Spúšťače (trigre) a udalosti (eventy).

Výpočtové úlohy v OpcDbGateway sa nazývajú [eventy](#). Každý event je štartovaný svojim [trigrom](#). Trigger môže štartovať jeden alebo viac eventov. Event má vždy len jeden trigger. Pre eventy je konfigurovaná ich priorita.

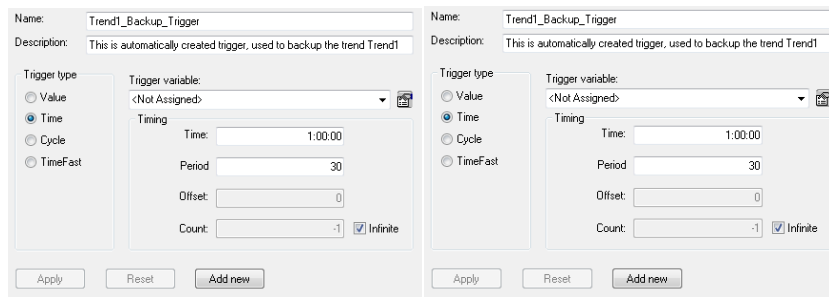
Pri vzniku eventu je tento vložený do [jednej z dvoch prioritných front – synchronnej alebo asynchronnej](#), ktoré sú využívané na vykonávanie asynchronných alebo synchronných výpočtov. Eventy zo synchronnej fronty sú vykonávané podľa priority eventov uložených v synchronnej fronte v synchronnom výpočtovom vlákne a eventy z asynchronnej fronty sú vykonávané podľa priority eventov v asynchronnej fronte v asynchronnom výpočtovom vlákne. Vykonané eventy sú z front okamžite odstraňované. Prepínanie výpočtových vlákien je rovnomerné. Vyhodnocovanie priority v týchto dvoch frontách je nezávislé. Dôsledkom tejto funkcionality je to, že event s nižšou prioritou v asynchronnej fronte môže byť vykonaný aj skôr ako event s nižšou prioritou v synchronnej fronte.

Trigre sú [časové](#) alebo [hodnotové](#) (Obrázok 3). Časové trigre sú buď [jednorazové](#) alebo [cyklické](#). U cyklických trigrov (v závislosti na type) je možné definovať počiatok vykonávania ako aj počet cyklov, ktoré sa majú vykonať.

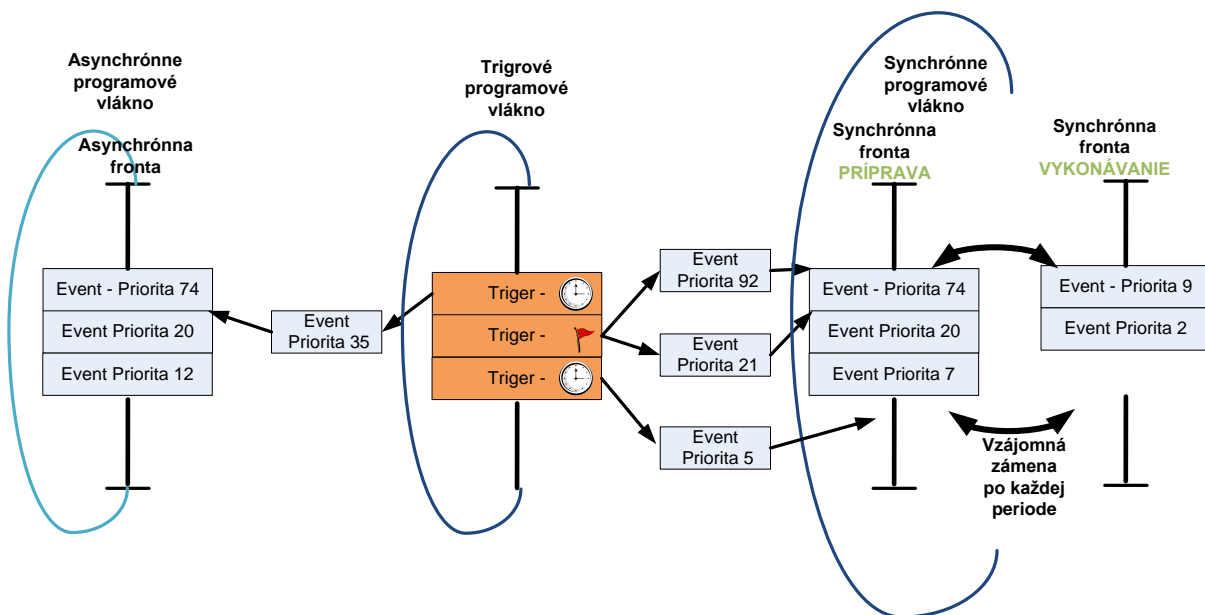
[Vyhodnocovanie trigrov](#), na základe ktorého sa vykonáva vkladanie eventov do vyššie spomenutých front, [prebieha v osobitnom výpočtovom vlákne](#) (Obrázok 4). Pri vyhodnotení trigra, ktorý spúšťa event vykonávaný v asynchronnom vlákne je event uložený do asynchronnej fronty podľa svojej priority. Ak by jeho priorita (v asynchronnej fronte) bola najvyššia, môže byť dokonca vykonaný okamžite a následne z fronty odstránený. [Synchronne fronty sú v sku-](#)

točnosti dve – prípravná a vykonávacia. Vyhodnotenie triga spôsobí, že sa event vloží do prípravnej synchronnej fronty. Na začiatku novej periódy synchronných výpočtov si prípravná a vykonávacia fronta vymenia úlohy. To má nasledovné dôsledky:

- V priebehu synchronnej periódy sa poradie eventov, ktoré boli uložené do synchronnej aktuálne výkonnej fronty nemení
- Keďže si fronty (výkonná a prípravná) vzájomne vymieňajú úlohy, nevykonané eventy z predchádzajúcej periódy zostanú na vykonanie v perióde nasledujúcej. Dlhodobé narastanie počtu eventov v synchronnej fronte signalizuje problém v konfigurácii alebo v dostupných výpočtových zdrojoch. Integrátor môže počet eventov sledovať a prípadne riešiť.



Obrázok 3 Dialogy pre konfigurovanie trigrov a eventov



Obrázok 4 Vykonávanie eventov v synchronnom a asynchronnom vlákne

Eventy ovplyvňujú nielen konfigurovateľnú výpočtovú funkcionálnu (*call function block*) ale tiež priamo zabezpečujú parametrizovateľné spúšťanie externých programov (*call ext. program*) vykonávanie SQL dotazov na procesných databázach (*do query, create report, sync.generated reports table*), generovanie log súborov a zápis parametrizovateľných užívateľských správ do nich (*make new alarm log file, make new log, write message to log*)

Funkčné bloky a príkazy

Výpočtové aktivity môžu byť štruktúrované do funkčných blokov. V rámci funkčných blokov sa vykonávajú príkazy. Funkčné bloky nie sú parametrizovateľné na rozdiel od príkazov, ktoré môžu mať maximálne jeden výstupný parameter a dva vstupné parametre (Obrázok 8).

Postupnosť vykonávania príkazov v rámci funkčného bloku je daná ich očíslovaním. Pri vykonávaní funkčného bloku sa príkazy vykonávajú vo vzostupnom poradí. V číslovaní môžu byť medzery. Existuje príkaz *CALL*, ktorý umožňuje volať iný funkčný blok. Keďže vykonanie tohto príkazu môže byť podmienené vstupným parametrom, je možné týmto spôsobom zabezpečiť vetvenia programu. K ďalšiemu vysvetleniu príkazov sa vrátíme po objasnení niekoľkých ďalších pojmov.

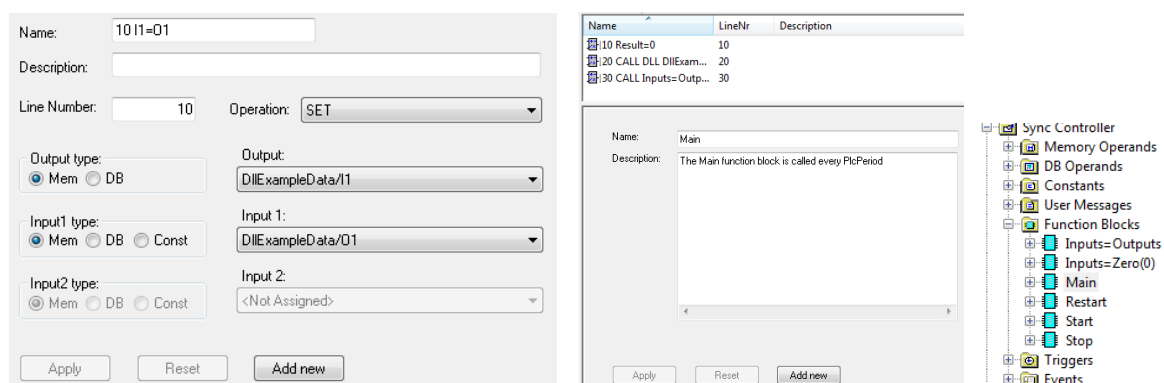
Existujú štyri preddefinované funkčné bloky *MAIN*, *START*, *RESTART* a *STOP* ktoré sa vykonávajú v každej integrovanej aplikácii. Tieto môžu a nemusia byť naplnené príkazmi. Funkčný blok *START* sa vykoná jediný krát a môže byť využitý na inicializáciu aplikácie. *OpcDbGateway* má implementovaný jednoduchý mechanizmus perzistencie dát. Ak chceme neskôr využiť dáta uložené pri ukončení aplikácie pri reštarte výkonnej aplikácie *OpcDbGateway*, potom je možné nakonfigurovať aby sa namiesto funkčného bloku *START* na začiatku spustil funkčný blok *RESTART*. Blok *STOP* sa vykoná jediný krát pri zastavení aplikácie.

Funkčný blok *MAIN* sa vykonáva cyklicky v synchrónnom programovom vlákne.

Funkčné bloky môžu byť volané ako eventy v synchrónnom alebo asynchrónnom programovom vlákne. Program *MAIN* sa v rámci synchrónneho programového vlákna vykonáva zvyčajne až po vykonaní funkčných blokov volaných ako eventy. Ak však takto volaných funkčných blokov bolo toľko že v danom synchrónnom cykle nemohol byť *MAIN* vykonaný, potom sa v nasledujúcom cykle volá ako prvý.

Programové konštrukcie dostupné pomocou funkčných blokov a príkazov nie sú také komfortné ako u programovacích jazykov vysokej úrovne. Umožňujú však jednoducho nakonfigurovať (nie programovať) viaceré typické úlohy zberu, spracovania dát a sprostredkovania ich prenosu medzi externými zariadenia. Náročnejšie úlohy spracovania dát je možné naprogramovať vysokoúrovňovými programovacími jazykmi a integrovať do aplikácie riadenej cez *OpcDbGateway* pomocou rozširovacích *DLL*.

Rozširovaciu *DLL* je možné využiť vo funkčnom bloku pomocou príkazu *CALL DLL*.

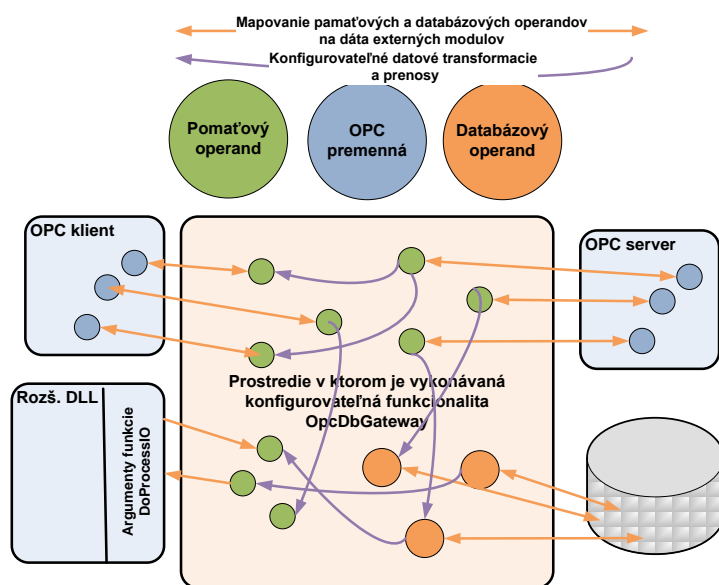


Obrázok 5 Konfigurovanie príkazov a funkčných blokov

Pamäťové a databázové operandy

Oblasť pamäťových operandov je miestom, kde dochádza k výmene informácií medzi rôznymi externými zariadeniami, databázami, výpočtovými modulmi implementovanými pomocou rozširovacích DLL (Obrázok 6).

Príkazy sa vykonávajú nad pamäťovými a databázovými operandami (budú popísané ďalej). Predstavujú súvislú neštruktúrovanú oblasť pamäti. Integrátor má možnosť vybrať si meno pamäťového operandu a tiež jeho umiestnenie pomocou jeho očíslovania. Pri konfigurovaní je možné vytvárať adresáre pamäťových operandov. Vytvorenie adresára však neovplyvňuje pozíciu pamäťového operandu v pamäti. – tá je daná iba ich poradovým číslom. .



Obrázok 6 Výmena a spracovanie dát medzi externými zariadeniami, výpočtovými modulmi a databázami

Dáta z externých zariadení sú mapované do oblasti pamäťových operandov. Pomocou konfigurovateľnej funkcionality *OpcDbGateway* pracujeme s pamäťovými operandami a zabezpečujeme spracovanie a výmenu dát medzi externými zariadeniami, databázami, programovými modulmi a samotným výpočtovým strojom *OpcDbGateway*.

Existuje jedna špeciálna oblasť pamäťových operandov s jednoznačne definovaným významom, ktorá slúži ako rozhranie medzi výpočtovým strojom *OpcDbGateway* a aplikáciou - **systémové operandy**. Prostredníctvom nich aplikácia má k dispozícii nielen informácie o stave výpočtového stroja ale tiež možnosť ovplyvniť jeho funkcionálnu.

U pamäťových operandov je možné vybrať si ich dátový typ (Obrázok 7), čo redukuje nutnosť implicitných transformácií počas výpočtov a teda ich vyššiu efektívnosť. Môžu byť nasledovných dátových typov: *bool*, *byte*, *currency*, *date*, *double*, *dword*, *float*, *integer*, *long*, *short*, *string*, *word*. Ak zvolíme pri definícii pamäťového operandu dátový typ *bool* **potom je možné definovať pamäťový operand ako jeden bit 16-bitového slova**, ktoré dátový typ *bool* používa pre prenos informácií.

Pamäťové operandy môžu slúžiť ako zdroj pre generovanie alarmu alebo zdroj pre potvrdzovanie alarmu ako bude ukázané ďalej. Pamäťovému operandu môže byť preto priradený alarm.

The image shows two side-by-side dialog boxes for configuring operands. The left dialog is for a memory operand named 'PlcSystemTime'. It has a description 'The current Plc system time', a system variable 'PlcSystemTime', a data type of 'DATE', and a memory address of '12'. There is also a 'Bit' checkbox set to '0' and a 'Digital alarm' dropdown set to '<Not Assigned>'. The right dialog is for a database operand named 'DBOperand1'. It has an empty description, a type of 'Column', a table of 'Trend1', and a field of 'ID'. The 'Row select through' is set to 'Row', and the 'Memory operands' dropdown is set to '<Not Assigned>'. Both dialogs have 'Apply', 'Reset', and 'Add new' buttons.

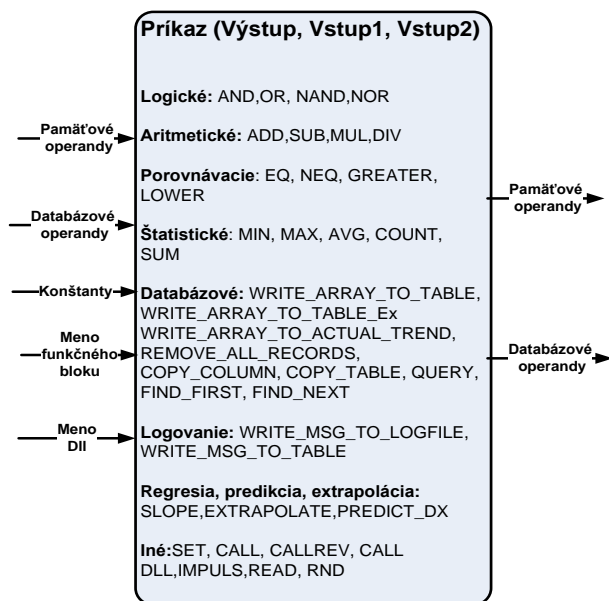
Obrázok 7 Dialog pre konfigurovanie pamäťových a databázových operandov

Databázové operandy mapujú obraz bunky, alebo stĺpca v definovanej tabuľke definovanej procesnej databázy. V istom zmysle môžu mať charakter indexovanej premennej. Ako index môže byť použitý vybraný pamäťový operand. Zmenou hodnoty tohto operandu je možné za behu meniť bunku v databáze ku ktorej databázový operand v danom okamihu pristupuje.

Databázové operandy umožňujú konfigurovateľný prenos dát medzi pamäťovými operandami a nimi samotnými. Existuje napr. možnosť kopírovať hodnotu premennej z pamäťového operandu do databázového operandu. Môžu byť použité ako argument v konfigurovateľnom príkaze (Obrázok 8).

Databázové operandy sú dôležitou súčasťou synchronného spracovania dát v OpcDbGateway. V každom synchronnom cykle sa aktuálne hodnoty databázových operandov aktualizujú z danej procesnej databázy a po vykonaní príslušných výpočtov sa zase prenášajú ich hodnoty automaticky do databázy. To, či je tento spôsob práce s databázou pre danú aplikáciu efektívny, je na uvážení integrátora, pretože príliš časté prístupy do databázy (v každom synchronnom cykle) môžu spôsobovať problémy. V takomto prípade je vhodnejšie použiť asynchrónne prístupy na databázu v čase, keď je to naozaj potrebné. Pre tento účel slúži niekoľko príkazov, ktoré využívajú pre prístup k databáze pamäťové operandy. Veľmi flexibilný je konfigurovateľný parametrizovateľný príkaz ktorým je možné vykonať SQL dotaz nad databázou, a ktorý umožňuje ako parametre použiť pamäťové operandy, a teda za behu zmenou hodnoty týchto operandov meniť jeho účinok. Pomocou konfigurovateľného SQL príkazu je tiež možné spúšťať na databáze uložené procedúry.

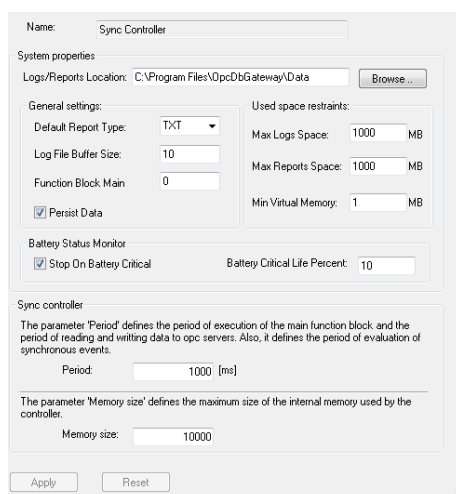
Keďže niektoré konfigurovateľné príkazy používajú ako parameter databázový operand, potom v prípade že ich chceme využívať, použitiu databázových operandov (a teda aj zapojeniu ich implicitnej synchronnej aktualizácie) sa nevyhneme. Avšak aj tieto príkazy dokážeme zvyčajne nahradiť príkazom pre volanie parametrizovateľných SQL príkazov. Najflexibilnejším spôsobom práce s databázami z OpcDbGateway je naprogramovanie pre danú aplikáciu špecifickej databázovej funkcionality v rámci rozširovacej DLL. Samozrejme, za túto flexibilitu platíme vyššou prácnosťou.



Obrázok 8 Konfigurovateľné príkazy - prehľad

Funkcionalita výkonnej aplikácie OpcDbGateway

Výkonná aplikácia OpcDbGateway je implementovaná ako Windows – program bez vlastného užívateľského rozhrania. Ako už bolo spomenuté, je to vlastne OPC server zabezpečujúci prístup k hodnotám svojich pamäťových operandov. V dôsledku toho, pre spustenie výkonnej aplikácie je potrebné spustiť aspoň jednu OPC klientsku aplikáciu, ktorá tento OPC server výkonnej aplikácie naštartuje. Takouto klientskou aplikáciou môže byť napr. jednoduchý OPC klient s vlastným užívateľským rozhraním sprostredkujúci prístup k hoci jedinému pamäťovému operandu alebo úplná vizualizačná aplikácia. Konfiguračný program obsahuje tiež OPC klienta, ktorý sa dá dobre využiť pre spustenie výkonnej aplikácie a jej ladenie. Pomocou neho je napr. možné meniť hodnoty systémových premenných a ovplyvňovať tak beh OpcDbGateway. Synchronne operácie sú vykonávané pomocou synchronného kontroléra (Obrázok 9). Períodu cyklu tohto kontroléra je možné parametrizovať v milisekundách.



Obrázok 9 Konfigurácia funkcionality synchronného kontroléra

OpcDbGateway obsahuje vlastného OPC klienta ktorý umožňuje komunikovať s externými zariadeniami prostredníctvom externých OPC serverov. OPC položky z týchto OPC serverov môžu byť mapované na pamäťové operandy a prostredníctvom týchto byť sprístupnené pre interný OPC server, iné externé OPC servery alebo pre rozširovacie DLL.

Name	OPC Address	DataType	Direction	memory address
Increment.Byte	Increment.Byte	BYTE	input/output	1001
Increment.Currency	Increment.Currency	CURRENCY	input/output	1002
Increment.Date	Increment.Date	DATE	input/output	1003
Increment.Double	Increment.Double	DOUBLE	input/output	1004
Increment.Dword	Increment.Dword	DWORD	input/output	1005
Increment.Float	Increment.Float	FLOAT	input/output	1006

OPC Server

You can select the parameter 'ProgID' using the following button or you can write this parameter also manually in one of the following format: a) ProgID b) Computer Name:ProgID or c) IP:Address:ProgID

ProgID:

Description:

Check connection & Re-connect of OPC Server

The parameter 'Period' defines the period of checking of the connection to the external OPC Server and the period when the OpcDbGateway attempts to establish the lost connection again (re-connection).

Period [60-3600]: s

Asynchronous update

With this option you can specify whether data will be readed automatically or when used. If this option is disabled you MUST use any OPC item in any function block.

Update rate: ms

Read opc items from the server's cache

The scheduler reads opc items from the DEVICE of the opc server. There are no active opc items in the server. The scheduler controls when and what is read from the connected device.

Obrázok 10 Konfigurácia OPC klienta pre prístup k externému OPC serveru

V rámci cyklu synchronného kontroléra dochádza k automatickému cyklickému načítaniu OPC položiek z externých OPC serverov a aktualizácii na ne mapovaných pamäťových operandov. Aby k tomu naozaj dochádzalo pre daný OPC server automaticky, je potrebné aby mapovaný pamäťový operand bol použitý aspoň v jednom funkčnom bloku.

Množstvo OPC premenných, ktoré majú byť aktualizované v jednom cykle môže byť veľmi veľké. V takomto cykle sa aktualizujú aj také premenné kde vôbec nedošlo k zmene. Existujú dve možnosti ako aktualizovať mapované operandy na začiatku každého synchronného cyklu. Buď spomínané **synchronne vyčítavanie položiek z externého OPC servera**, alebo **aktualizácia pamäťových operandov na základe hodnôt OPC premenných vo vyrovnávacej pamäti interného OPC klienta**. Pri synchronnom vyčítavaní máme opäť dve možnosti - buď vyčítavať z vyrovnávacej pamäti externého OPC servera alebo z hodnôt na pripojenom zariadení. Ktorá z týchto možností je vhodnejšia závisí na type pripojeného zariadenia a komunikačného protokolu medzi nim a externým OPC serverom.

Vyčítavanie z vyrovnávacej pamäti OPC klienta bude zvyčajne podstatne efektívnejšie. OPC server notifikuje interného OPC klienta o každej zmene OPC premennej. Maximálnu frekvenciu týchto notifikácií posielaných na OPC klienta z externého OPC servera je možné nastaviť nezávisle na perióde synchronného cyklu OpcDbGateway.

K internému OPC klientovi môže byť pripojených viacero OPC serverov. Bežná situácia, kedy externé zariadenie sa odpojí a prestane fungovať jeho OPC server nesmie ohroziť funkcionality integrovanej aplikácie ako celku a musí byť možnosť za behu aplikácie OPC server

znovu pripojiť. Z uvedeného dôvodu je možné parametrizovať s akou periódou sa má pokúšať OPC klient pripojiť na externý OPC server v prípade jeho odpojenia.

OPC klient a interný OPC Server využívajú vlastné programové vlákna v podstate nezávislé na vyššie spomenutých troch programových vláknach synchrónnom asynchrónnom a tigrovom.

Spôsob vyčítavania dát z pamäťových operandov zabudovaným OPC serverom je vlastne daný, okrem jeho možností, hlavne parametrizáciou externých OPC klientskych aplikácií.

Výpočtový stroj má zabudovaných niekoľko štandardných funkcionalít zabezpečovaných SCADA systémami ako napr.: **alarmový systém, systém generovania tlače a správy reportov, stráženie vlastného aplikačného prostredia a využívaných zdrojov ako napr. stavu batérie alebo záskokového zdroja, využitej virtuálnej pamäte, priestoru na disku využívaného procesnými databázami, log-súbormi a reportmi.** Pre komunikáciu s nimi využíva OPC premenné interného OPC servera namapované na príslušné pamäťové operandy zo systémovej oblasti. Pomocou systémovej premennej je dokonca možné zastaviť alebo spustiť beh synchrónneho kontroléra. Vizualizačné aplikácie spolupracujúce s OpcDbGateway môžu byť pripojené pomocou komunikačných štandardov OPC DA, OPC XML DA, OPC UA. Tieto možnosti pripojenia sa dajú rozšíriť o ďalšie komunikačné protokoly implementované pomocou rozširovacích DLL. Takýmto spôsobom je napr. možné zabezpečiť komunikáciu cez DDE protokol.

Alarmový systém je možné realizovať dvomi spôsobmi – využitím štandardného OPC AE rozhrania alebo ako špeciálny alarmový systém prevádzkovaný výpočtovým strojom OpcDbGateway. Tento systém umožňuje viac zdrojové ovplyvňovanie generovania alarmu ako aj viac zdrojové potvrdzovanie alarmu. Okrem stavového automatu pre spracovanie alarmov obsahuje zabudovaný alarmový systém prehľadnú tabuľku stavov alarmových zdrojov ako aj tabuľku alarm-logu uloženú v procesnej databáze. Keďže zdrojom alarmu ako aj jeho potvrdzovania je pamäťový operand, môže nastavenie alebo potvrdzovanie alarmu byť vykonávané pomocou ľubovoľného externého zariadenia alebo aplikácie pripojenej k OpcDbGateway.

Funkcionalita aktuálnych trendov predstavuje tabuľky s pevne definovaným počtom riadkov kde najnovší záznam sa ukladá na začiatok tabuľky a ostatné sa posúvajú o jednu pozíciu nadol. Vizualizačná aplikácia pripojená priamo k tabuľke procesnej databázy môže zobrazovať jeden alebo viac posúvajúcich sa grafov generovaných na základe údajov z týchto tabuliek.

Funkcionalita historických trendov zabezpečuje periodické ukladanie hodnoty jednej alebo viacerých premenných do tabuľky s postupne narastajúcim počtom riadkov. Je možné tiež nastaviť periodické zálohovanie a vymazávanie uvedenej tabuľky. Táto funkcionalita je spúšťaná cyklickým asynchrónnym trigrom a teda nevyžaduje beh synchrónneho kontroléra. Tvorba historických trendov je podporovaná softvérovým pomocníkom, ktorý zabezpečí automatické vygenerovanie zodpovedajúcej zostavy trigrov a eventov.

Systém užívateľských a alarmových správ je viacjazyčný. Pomocou systémovej premennej je možné prepnúť v akom jazyku sa budú nakonfigurované správy ukladať buď do databázy alebo do súborov. Ukladané správy môžu obsahovať maximálne päť parametrov, ktorých náhradou sa zobrazia aktuálne hodnoty k nim priradených pamäťových operandov.

Logovanie činnosti systému s parametrizovateľnou detailnosťou je v anglickom jazyku a umožňuje rozlíšiť ktoré správy sú informačné, ktoré varovné a ktoré chybové.

Štartovanie externých programov je taktiež parametrizovateľné čo zabezpečuje možnosť parametre nahradiť aktuálnym obsahom pamäťových operandov.

Metodika návrhu integrovanej aplikácie

1. Definovanie cieľov aplikácie
2. Výber externých zariadení, programov a databáz a rozširovacích DLL z ktorých dáta majú byť spracovávané
3. Vytvorenie vonkajšieho dátového modelu aplikácie – rozhrania medzi externými entitami a OpcDbGateway
4. Rozdelenie funkcionality na synchrónne a asynchrónne úlohy
5. Návrh funkčného modelu – funkčných blokov
6. Návrh využitia štandardných komponentov systému ako napr. alarmový systém, historické trendy...
7. Návrh ostatnej funkcionality realizovanej funkčnými blokmi
8. Návrh interného dátového modelu – pamäťových a databázových operandov pre zabezpečenie funkcionality podľa bodu 7.
9. Implementáciu nových rozširovacích DLL
10. Návrh uložených procedúr na databázach
11. Návrh vizualizácie prípadne iného spôsobu štartovania OpcDbGateway, včítane náväznosti vizualizácie na štandardnú funkcionality (alarmy, historické a aktuálne trendy...)

Záver

Možnosti, ktoré poskytuje OpcDbGateway sú naozaj rozsiahle. Umožňuje integrovať aplikácie, ktoré spracúvajú veľké množstvo dát z externých zariadení pri periódach synchrónneho kontroléra zvyčajne rádovo sekundy, a tiež štartovať externé rozširovacie moduly, externé programy, skripty a na databázach uložené procedúry s periódami nižšími a s veľmi rýchlymi výpočtovými aktivitami.

Fakt, že okrem štandardných OPC technológií umožňuje využívať tiež webové služby či už implementované podľa OPC XML DA a OPC UA štandardov alebo v rámci rozširovacích DLL, umožňuje využiť ho v rozsiahlych systémoch s SOA (na služby orientovaná architektúra). Jeho využitie je efektívne aj v menších aplikáciách kde sa jedná o sprostredkovanie komunikácie medzi dvoma externými zariadeniami, alebo medzi zariadeniami a databázou.

Veľkou výhodou je jeho rozširovateľnosť konfiguračnej funkcionality. Integrátorské firmy, ktoré ho využívajú si môžu vytvoriť z neho systém pokrývajúci ich špecifické potreby.