

SMARTSCADA PROGRAMSKI ALAT ZA KREIRANJE UPRAVLJAČKOG I NADZORNOG SOFTVERA

Jovan Mirilović dipl. Ing, mr Velibor Ilić,

TechnoSoft, e-mail: jmiril@EUnet.yu, ilicv@EUnet.yu, http://www.TehnoSoft.co.yu

Apstrakt: U radu se govori o postupku izrade nadzornih i upravljačkih programa upotrebom SmartSCADA programskog alata; sa primerom primene u NIS Naftagas NOVI SAD.

Ključne reči: procesno upravljanje, nadzorni sistemi, OOP (Object Oriented Programming), OPC (Ole for Process Control), ActiveX, baze podataka, client/server, Delphi.

SmartSCADA PROGRAMMING TOOL FOR DEVELOPMENT CONTROL AND MONITORING SOFTWARE

Abstract: This text describes SmartSCADA programming tool for development control and monitoring software.

Keywords: process control, data acquisition, OOP (Object Oriented Programming), OPC (Ole for Process Control), ActiveX, database, client/server, Delphi.

1. UVOD

Razvoj savremenih objektno orijentisanih programskih jezika u WINDOWS okruženju, sa mogućnošću vizuelnog programiranja pomoću komponenti, omogućio je enormno ubrzanje u kreiranju programa raznih namena. Naročito se razvio deo koji se odnosi na upravljanje bazama podataka. Danas se na tržištu može naći više softverskih paketa koji pružaju mogućnost vizuelnog razvoja aplikacija kao što su Borland Delphi, Borland C++ Builder, Microsoft Visual Studio (Visual C++, Visual Basic), Microsoft Access i drugi.

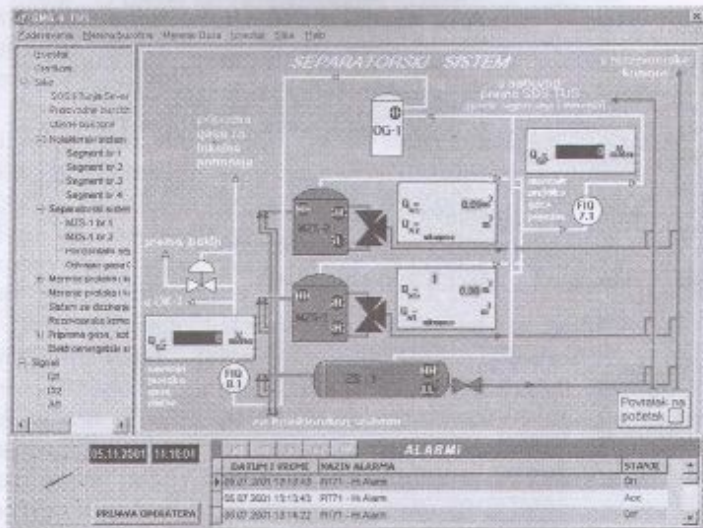
Ovi programski paketi se odlikuju velikom brzinom izrade aplikacija (RAD, Rapid Application Develop). Programi se u njima kreiraju postavljanjem i povezivanjem softverskih komponenti u celinu, poput slaganja LEGO kockica. Da bi se moglo na ovaj način programirati, ovi programski paketi moraju da imaju niz predefinisanih komponenti, formi i čarobnjaka (wizard) za razne delove programa.

Danas na tržištu postoji veliki broj nezavisnih proizvođača softverskih komponenti za razne namene, pa tako i za razvoj SCADA aplikacija. Međutim, većina njih obuhvata samo pojedine elemente jednog SCADA sistema.

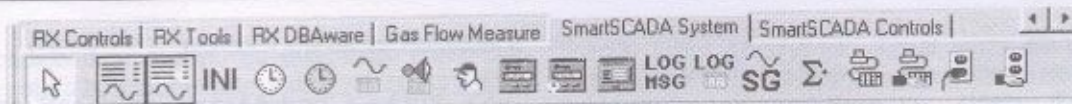
SmartSCADA predstavlja nadogradnju (sistem softverskih komponenti) za razvojno okruženje DELPHI i C++ Builder, kompanije Borland, koje omogućava ovim razvojnim sistemima da postanu sistemi za razvoj SCADA aplikacija, zadržavajući jednostavnost i brzinu kreiranja aplikacija.

Sistem komponenti, SmartSCADA, smo kreirali iz sledećih razloga:

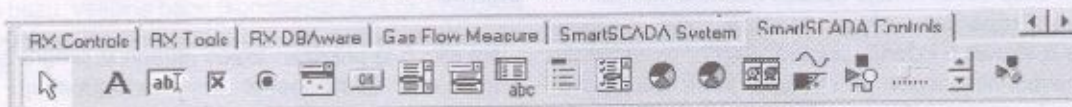
- Brzina izrade aplikacije. Uvek postoji vremensko ograničenje, odnosno rok do kada aplikacija mora da se završi i proveriti u praksi.
- Veliki dijapazon zahteva korisnika aplikacija.
- Nedovoljna definisanost projekata i često nejasni ciljevi. Što kasnije dovodi do izmena u softveru, pa i hardveru.
- Primena standardnih objektno orijentisanih programskih jezika, DELPHI ili C++ Builder.
- Mogućnost izrade aplikacija sa minimumom koda, čime se drastično povećava stabilnost softvera.
- Standardan interfejs za komunikaciju prema



Slika 1 – Izgled jedne gotove aplikacije na objektu NIS-NAFTAGAS-a



Slika 2 - Paleta SmartSCADA System



Slika 3 - Paleta SmartSCADA Signal Controls

različitim hardverskim uređajima i raznim softverima.

Niža cena krajnje aplikacije jer se ne koristi Run Time modul, već je proizvod aplikacija u izvršnom kodu (EXE).

2. ORGANIZACIJA SMARTSCADE

Zahtevi nadzorno/upravljačkih sistema koje pokriva SmartSCADA su:

- Komuniciranje sa signalima iz realnog sveta.
- Alarmiranje nedozvoljenih stanja signala.
- Konverzija vrednosti signala iz sirovih u inženjerske i obrnuto.
- Arhiviranje podataka stanja signala.
- Prikaz trenutnih vrednosti signala na različite načine.
- Prikaz arhiviranih vrednosti signala na različite načine.
- Menjanje vrednosti signala, radi upravljanja procesima.
- Kreiranje raznih izveštaja, kako standardnih tako i nestandardnih.

Komponente koje se upotrebljavaju u SmartSCADI, mogu se podeliti na vizuelne (preko kojih se prikazuje situacija nadgledanog sistema MMI) i nevizuelne komponente koje omogućavaju očitavanje i obradu signala iz realnog sveta.

SmartSCADA komponente su dizajnirane tako da se određene metode objekata automatski izvršavaju bez posebnog programiranja. Na primer, komponenta TAlarmHandler konstantno nadgleda signale koji se nalaze u sistemu i ukoliko neki od signala prekorači dozvoljeni opseg vrednosti, komponenta automatski prikazuje na ekranu poruku o alarmu sa pratećim zvučnim signalom i u bazu podataka upisuje datum/vreme pojave

alarma, opis i nivo alarma.

Prenošenje stanja nekog signala na bilo koju vizuelnu ili nevizuelnu komponentu se obavlja prostim povezivanjem osobina jedne komponente sa signalom, što se ostvaruje u vreme dizajniranja aplikacije.

3. KOMPONENTE SMARTSCADE

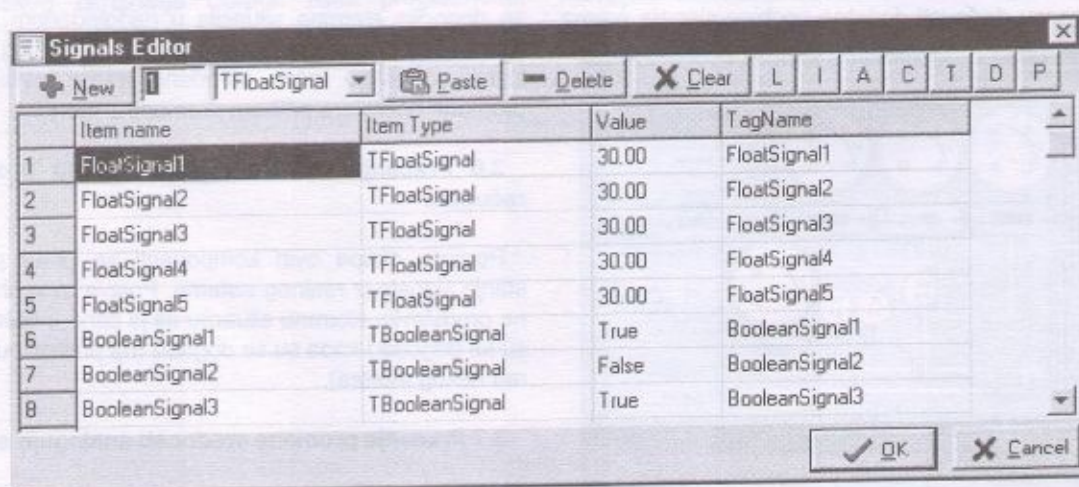
Komponente SmartSCADE su podeljene na dve palete komponenti, SmartSCADA System na kojoj se nalaze sistemske nevizuelne komponente, i SmartSCADA Signal Controls na kojoj se nalaze vizuelne komponente.

3.1 DEKLARACIJA I OBRADA SIGNALA

Osnovu sistema SmartSCADA predstavljaju nevizuelne komponente, signali različitog tipa (TFloatSignal, TBooleanSignal, TIntegerSignal, TDateTimeSignal i TStringSignal). Upravljanje signalima se vrši pomoću komponente TSignalCollection. Komponenta omogućava kreiranje, brisanje i podešavanje signala.

Signalne komponente su izrađene u skladu sa standardom OPC DA 2.04. Osnovne osobine (property) signala su:

- TagName: tehnološki naziv.
- Description: opis osnovnih funkcija.
- AdditionalProperties: dodatni skup osobina koje korisnik može naknadno definisati.
- IniFile: određuje u koji ini fajl će biti snimljene osobine signala.
- TimeStamp: opisuje vreme između dva osvežavanja



Slika 4 - Editor signala

stanja signala.

- RawDataType: tip sirovih podataka koji se dobijaju sa upravljačkih sistema.

- Alarms: skup osobina koje opisuju alarmna stanja.
- ConvertProperties: skup osobina koje opisuju konverziju iz sirovih u inženjerske jedinice i obrnuto.

Osim navedenih osnovnih osobina svaki od tipova signala ima i svoje specifične osobine.

Osobine signala se mogu obrađivati i menjati pomoću standardnih editora (ObjectInspector) ili specijalizovanih editora kako za vreme dizajniranja, tako i za vreme izvršavanja izrađenih aplikacija.

Specijalizovani editor se koristi prilikom izrade programa (dizajniranja) i omogućavaju grupno podešavanje signala u aplikaciji. Kreirani su sledeći editor:

Signals Editor - omogućava kreiranje, brisanje i konfigurisanje signala koji se upotrebljavaju u sistemu. Editori koji se opisuju u ostatku teksta pozivaju se pomoću dugmića koji se mogu videti na gornjem delu editora.

Link editor – prikazuje listu svih signala i komponenti na koje se ti signali mogu vezati. Omogućava povezivanje komponenti upotrebom drag and drop tehnike (prevlačenjem mišem).

Ini editor – povezuje signale sa ini komponentom i na taj način se vrednost signala automatski snima i učitava iz ini fajla. Kada je signal povezan na Ini komponentu prilikom učitavanja programa dobija vrednost kakvu je imao pre zatvaranja aplikacije.

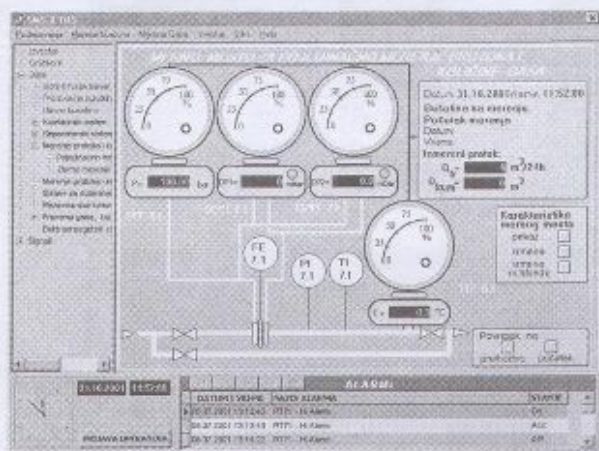
Alarm Editor – omogućava grupno podešavanje alarma (alarmnih nivoa, alarmnih poruka, zvukova koji se javljaju u slučaju alarmnih situacija i prioritet alarma).

Convert Editor – pomoću ovog editora se podešava tip i parametri konverzije signala (sirov signal u inženjerske jedinice i obrnuto).

Active Editor – u ovom editoru se podešava stanje signala (da li su aktivni ili ne).

Description Editor – omogućava upisivanje komentara za svaki signal (funkcija signala, tip, gde je povezan u realnom sistemu i drugi bitni komentari vezani za određeni signal).

Additional Properti Editor – predstavlja editor pomoću koga se mogu dodavati dodatne osobine signala prema



Slika 5.

specifičnim zahtevima korisnika.

3.2 Prikaz i podešavanje vrednosti analognih signala

Na slici 5 je prikazan izgled ekrana iz programa gde su primenjene komponente koje omogućavaju prikaz i promenu vrednosti signala. Na raspolaganju su komponente u obliku prekidača, klizača, progres barova i mernih instrumenata.

3.3 Vizualizacija procesa

Grupa komponentata koja omogućava vizuelno predstavljanje situacije u realnom procesu, ima sposobnosti za:

- Promenu boje objekta u nekom geometrijskom obliku (četvorougao, krug, i sl.) u zavisnosti od stanja signala.
- Smenjivanje slika u zavisnosti od stanja signala.
- Pokretanje i zaustavljanje animacija.

3.4 Alarmni sistem

Komponente koje omogućavaju detektovanje i evidenciju alarmnih situacija u sistemu kojim se upravlja. Realizovana komponenta detektuje promene digitalnih signala i analognih signala u željenom opsegu (sa razlikovanjem gornje i donje granice).

Kada dođe do alarmne situacije na ekranu se pojavljuje tekstualna poruka koja se beleži u bazu podataka, što kasnije omogućava generisanje izveštaja o pojavi alarmnih situacija u određenom vremenskom periodu.

3.5 Bezbednosne komponente

Sistem komponenti koji omogućava zaštitu pojedinih delova programa pomoću lozinki. Pored zaštitne uloge, sistem komponenti omogućava praćenje upotrebe procesne aplikacije (startovanje i zatvaranje programa, podešavanje parametara, generisanje izveštaja). Svaka promena operatera (unos nove lozinke) se beleži u bazu podataka tako da se lako može utvrditi koji operater je bio prisutan kada se vršila izmena parametara sistema ili kada se dogodila alarmna situacija u nadgledanom sistemu. Komponente se mogu podesiti tako da se dijalog za unos lozinke pojavi u tačno određeno vreme (evidentiranje promene operatera prilikom promene smene).

3.6 Praćenja eksternih događaja sa procesnog računara

Pomoću skupa ovih komponenti se prate promene stanja signala iz realnog sistema. Pojava ovakvih signala ne predstavlja alarmnu situaciju ali je bitno evidentirati da su se dogodili i kada su se dogodili (na primer, puštanje u rad nekog motora).

3.7 Praćenje promene vrednosti analognih signala

TSignalRecorder je komponenta na koju se vezuju analogni signali čije vrednosti želimo da pratimo radi

kasnije obrade. Ima osobinu da automatski kreira sva potrebna polja u bazi sa imenima analognih signala koji su vezani za nju. Može se podesiti period snimanja vrednosti signala u bazu, veličina baze (konstantan broj recorda), da li se popunjava sa test podacima za vreme dizajniranja programa.

3.8 Generisanje izveštaja

Izveštaji mogu biti grafički ili tabelarni, i uglavnom se generišu iz baza podataka. Izveštaji se mogu prilagoditi potrebama krajnjeg korisnika. Izveštaji se prikazuju u vremenskom periodu po izboru korisnika. Neki od mogućih izveštaja koji se mogu dobiti su:

- Izveštaj o alarmima.
- Izveštaj o događajima u realnom sistemu.
- Izveštaj o upotrebi procesne aplikacije (startovanje i izlasci iz programa i podešavanje osetljivih delova programa).

· Tabelarni izveštaj o promeni analognih signala (temperature, pritisci, nivoi).

Pored standardnih izveštaja moguće je kreirati specifične grafičke izveštaje kao što je:

- Grafički izveštaj koji vadi podatke iz baze AnalogRecorderTable, a može da prikazuje i trenutne vrednosti iz tih merenih veličina:

4. OTVORENOST SMARTSCADE

Signali u sistemu SmartSCADA sistemski su kreirani tako da sadrže editor kojim se omogućava podešavanje sistema bez izmena u izvornom kodu i kompajliranja krajnje aplikacije. Editor se poziva pomoću PopUp menija bilo koje od vizuelnih komponenti na koju je vezan dati signal. Omogućeno je grupno konfigurisanje više signala pozivanjem Edit metode u komponenti TSignalCollection. Naravno, programer koji kreira SCADA aplikaciju ima mogućnost da zaštiti šifrom ili potpuno onemogućiti konfigurisanje vrednosti programa u krajnjoj aplikaciji.

Editor je kreiran po uzoru na ObjectInspector iz programskog okruženja Delphi. Radi preglednosti, parametri koji se mogu podešavati su podeljeni na grupe (standardni, za konverziju, alarmni, dodatni). Ukoliko se

pozove editor za analogni signal može se prikazati i grafikon na kome se vidi promena vrednosti signala u vremenu.

5. ZAKLJUČAK

Sistem izrade SCADA softvera preko specijalizovanih objekata (komponenti) u standardnim objektno orijentisanim jezicima se, po lakoći upotrebe i brzini izrade softvera, približio gotovim paketima za izradu SCADA. Sve veći broj proizvođača procesnih (PLC) računara, isporučuje drajvere za OPC servere za komunikaciju sa svojim uređajima. Na taj način se sve više širi upotreba vrednost softverskog paketa za izradu SCADA softvera prikazanog u ovom radu.

LITERATURA

Mirilović, J., Ilić, V., (2001), "Izrada upravljačkog i nadzornog softvera upotrebom objektno orijentisanih programskih jezika", II Seminar Automatike, AUP 2000, Banja Koviljača.

"OPC/Open Control", OCSTechnical.pdf, <http://www.opctoolkit.com>,

"Ole for Process Control – Data Access Automation Interface Standard", Version 2.1, OPC_20_Automation_Interface.pdf, January 1999

Savić, D., (1992): "Uvod u objektno programiranje", PC Program, Beograd

Yao, P., Leinecker, C. R., (1995): "Visual C++ 5 Biblija", Mikro Knjiga, Beograd

Papeš, I., (1990): "Turbo Paskal 5.5", Naučna Knjiga, Beograd

Hansen, A., (1991): "Programiranje na jeziku C", Mikro Knjiga, Beograd

Stajić, D., Bilinski, P., (1991): "Računarske telekomunikacije i mreže", Tehnička Knjiga, Beograd

Stojić, M., (1978): "Kontinualni sistemi automatskog upravljanja"

Cantu, M., (2000): "Delphi 5, Detaljan Izvornik", Kompjuter biblioteka, Čačak

Borland, (1992): "Programers reference"

Borland, (1992): "Language guide"