

Andrzej KUSEK

ANALIZA MOŻLIWOŚCI PAKIETÓW WIZUALIZACYJNYCH TYPU SCADA

Streszczenie. W artykule przedstawiono wybrane problemy związane z wizualizacją procesów przemysłowych. Omówione zostały podstawowe możliwości pakietów wizualizacyjnych. Analizie została poddana struktura bazy danych zmiennych procesowych, sposób komunikacji ze sterownikami przemysłowymi oraz możliwości graficzne wybranych pakietów.

ANALYSIS OF SCADA PROGRAMS FEATURES

Summary. In paper some problems of industrial processes visualization are described. It presents basic features of SCADA programs. The paper contains also analysis of real-time database structure, the way of communication with programmable controllers and graphical editors in some programs.

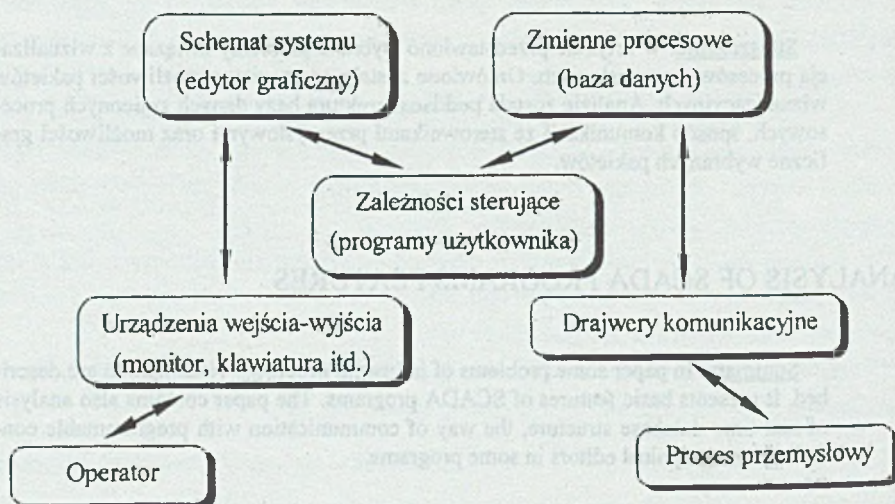
DIE ANALYSE DER MÖGLICHKEITEN VON SCADA PROGRAMMEN

Zusammenfassung. In diesen Artikel stellte man manche Probleme vor, die mit der grafischen Vorstellung der Industrie Processen verbunden sind. Es wurden die grundsetzlichen Möglichkeiten der SCADA Programme besprochen. Man analysierte die Strukturen von Database, die Methode der Kommunikation mit Hardware und grafische Möglichkeiten mancher Programme.

1. Wstęp

Niniejsze opracowanie zawiera przegląd możliwości programów do tworzenia aplikacji wizualizacyjnych. Analizie zostały poddane następujące programy:

- ◆ P1200, wersja dla DOS,
- ◆ P1200, wersja dla Windows, oba firmy CEGELEC,
- ◆ InTouch firmy Wonderware (Windows),
- ◆ ASIX firmy ASKOM (DOS),
- ◆ program powstający w firmie PROLOC (Windows).



Rys. 1. Struktura aplikacji wizualizacyjnej
Fig. 1. Structure of visualization application

Na rysunku 1 schematycznie została przedstawiona struktura aplikacji wizualizacyjnej. Jak widać, w jej skład wchodzi cztery zasadnicze części:

- ◆ schemat systemu tworzony za pomocą edytora graficznego,
- ◆ baza danych zmiennych procesowych,
- ◆ zależności sterujące,
- ◆ drajwery komunikacyjne.

W ogromnym uproszczeniu zadanie aplikacji wizualizacyjnej sprowadza się do przekazywania danych o stanie procesu przemysłowego do operatora oraz jego poleceń do poszczególnych urządzeń wykonawczych.

Kolejne części artykułu zawierają opis narzędzi do tworzenia wymienionych elementów w poszczególnych pakietach.

2. Baza danych zmiennych procesowych

Baza danych zmiennych procesowych jest jedną z zasadniczych części programów wizualizacyjnych. Zmienne procesowe pozwalają bezpośrednio komunikować się ze sterownikami przemysłowymi. Zmiana stanów odpowiednich obszarów pamięci sterownika poprzez zmianę wartości zmiennych procesowych pozwala informować użytkownika o przebiegu procesu przemysłowego. I odwrotnie - polecenia wydawane przez operatora oddziałują najpierw na wartości zmiennych procesowych, które z kolei wpływają na stan obiektów sterowanych przez aplikację.

Struktura baz danych zmiennych procesowych w analizowanych programach różni się nieco od siebie.

W przypadku najstarszego omawianego programu, P1200 w wersji dla DOS-a, zmienne występujące w programie są podzielone na 64-elementowe strony. Użytkownik dysponuje 64 stronami z elementami typu bitowego i tyłomaż stronami ze zmiennymi analogowymi. Oprócz tego występuje jeszcze podział na zmienne wewnętrzne programu i zmienne komunikujące się ze sprzętem. Deklaracja, czy dana strona jest wewnętrzna, czy zewnętrzna musi nastąpić na początku konfiguracji programu. Nie przewidziano symbolicznych nazw zmiennych - dostęp do nich następuje przez podanie numeru strony i numeru elementu na stronie. Oprócz zmiennych bitowych i analogowych występują jeszcze zmienne tekstowe, jednak ich liczba jest ograniczona do 64. Jak widać, struktura bazy danych w tym programie jest raczej prosta i jednocześnie mało wygodna w użyciu.

Zupełnie inną koncepcję struktury bazy danych zastosowano w następnej wersji P1200, tym razem w wersji dla Windows. Zrezygnowano z podziału zmiennych na strony. Zwiększona została także liczba typów zmiennych. Do dyspozycji są zmienne bitowe, tekstowe i analogowe o długości 8, 16 i 32 bitów. Ulepszono zostało zarządzanie bazą danych. Nazwy zmiennych mogą składać się z maksymalnie czterech członów, co pozwala tworzyć hierarchię w bazie danych. Możliwe są później operacje na całych „gałęziach”, tzn. na grupach zmiennych, które mają identyczne pierwsze człony nazwy. Oprócz tego można

zdefiniować domenę i naturę dla zmiennej lub grupy zmiennych, co pozwala szybciej orientować się, czego dotyczy dana zmienna i jakiej części systemu dotyczy.

W przypadku następnego omawianego produktu, programu InTouch, mamy również do czynienia z hierarchiczną strukturą bazy danych. Jest ona podobna do DOS-owego systemu katalogów. Zmienne są podzielone na grupy, z których każda może zawierać kolejne grupy zmiennych lub zmienne. Grupą bazową jest predefiniowana grupa o nazwie \$System, do której należą wszystkie pozostałe grupy. Wszystkie zmienne są rekordami o jednakowym zestawie pól informujących między innymi o typie zmiennej, czy zmienna jest alarmem, czy jest zmienną wewnętrzną lub zewnętrzną programu itp.

Program ASIX posiada jednopoziomową strukturę bazy danych. Zmienne definiowane są w pliku tekstowym typu ASCII. Kolejne linie pliku zawierają informacje o poszczególnych zmiennych. Podaje się tam takie parametry, jak:

- ◆ nazwa zmiennej,
- ◆ adres lokalizujący zmienną w sterowniku,
- ◆ nazwa kanału transmisji,
- ◆ częstość aktualizacji zmiennej,
- ◆ funkcja przeliczająca wartość pochodzącą ze sterownika na wartość używaną w systemie ASIX.

Zmienne mogą mieć określony jeden z następujących typów:

- ◆ WORD - liczba 16-bitowa bez znaku,
- ◆ INTEGER - liczba 16-bitowa ze znakiem,
- ◆ DWORD - liczba 32-bitowa bez znaku,
- ◆ LONG - liczba 32-bitowa ze znakiem,
- ◆ FLOAT - liczba zmiennoprzecinkowa.

Zestaw zmiennych procesowych musi zostać zdefiniowany przed rozpoczęciem pracy nad planszami wizualizacyjnymi. Zmienne te są bowiem częścią parametrów odnoszących się do obiektów na ekranie.

Jeśli chodzi o program powstający w firmie PROLOC, to tutaj również mamy do czynienia z definiowaniem zmiennych w plikach tekstowych. Jest to związane z jednoczesnym definiowaniem obiektów graficznych widocznych na ekranie. Jak widać, zmienne w tym programie mają w pełni obiektowy charakter. Nie występuje tutaj rozdzielanie zmiennych procesowych od elementów graficznych, lecz każdy obiekt w bazie danych zawiera informacje zarówno o wartości zmiennej (skąd powinien ją pobrać), jak i o sposobie przedstawienia jej na ekranie.

Dla użytkownika dostępnych jest kilka rodzajów obiektów - ma on do dyspozycji następującą grupę elementów:

- obiekty statyczne - linia, prostokąt, okrąg, rysunek w postaci mapy bitowej itd.,
- obiekty dynamiczne - zestaw jak wyżej, ale wygląd obiektów na ekranie determinowany jest wartością pojedynczego bitu, zestawu bitów, słowa lub bitu i słowa.

Takie podejście do organizacji struktury bazy danych sprawia, że jest ona bardziej spójna i łatwa do zarządzania.

Przy okazji omawiania struktury baz danych zawierających zmienne w poszczególnych programach chciałbym zwrócić uwagę na dość istotną kwestię. Producenci przeważnie nie ograniczają liczby zmiennych, jakie mogą być zdefiniowane w ich programach. Wydawałoby się więc, że za pomocą jednego komputera wizualizacyjnego można zautomatyzować pracę nawet bardzo dużego zakładu przemysłowego. W praktyce jednak liczba zmiennych możliwych do zdefiniowania w systemie jest ograniczona dwoma czynnikami. Pierwszy z nich to czas cyklu sieci łączącej sterowniki przemysłowe ze stacją wizualizacyjną. Przy bardzo dużej liczbie zmiennych okres odświeżania ich wartości jest również bardzo długi (rzędu kilkunastu lub kilkudziesięciu sekund), co może wykluczyć sens takiego rozwiązania. Okres ten wydłuża się także wraz ze wzrostem liczby sterowników połączonych razem w sieci.

Drugim ze wspomnianych czynników są możliwości graficzne komputera, który jest używany jako stacja wizualizacyjna. Na ekranie komputera mieści się bowiem ograniczona liczba elementów graficznych. Muszą one mieć taką wielkość, żeby były czytelne i nie utrudniały pracy operatorowi systemu. Przy zmianie plansz wizualizacyjnych większość obiektów na ekranie jest ponownie rysowana. Przy bardzo dużej liczbie elementów graficznych operacja ta może trwać bardzo długo. Tak więc można stwierdzić, że liczba zmiennych w systemie jest zawsze ograniczona. W zależności od specyfiki systemu silniej może działać ograniczenie związane z przepustowością sieci albo z możliwościami graficznymi stacji wizualizacyjnej. Oczywiście można powiedzieć, że przy użyciu sprzętu nowszego i szybszego ograniczenia te nie będą tak dokuczliwe. Jednak bez względu na stopień zaawansowania sprzętu użytego do wizualizacji czynniki ograniczające liczbę zmiennych w systemie będą istnieć.

3. Edytor graficzny

Edytor graficzny jest elementem, który ma bezpośredni wpływ na końcowy wygląd aplikacji. Jego ergonomia wpływa również mocno na czas tworzenia aplikacji, gdyż tzw. „interfejs użytkownika” jest przeważnie najbardziej czasochłonną częścią tworzenia wizualizacji.

Podobnie jak w przypadku zmiennych procesowych, omawianie rozpocznie od najstarszego programu, P1200 dla DOS-a. O edytorze graficznym dostarczanym z tym produktem nie można powiedzieć wiele dobrego. Użytkownik dysponuje zestawem podstawowych obiektów, takich jak: linia, okrąg, prostokąt itd., które może umieścić na ekranie. Jednak po narysowaniu obiektu te tracą swą „obiektość”, tzn. cały rysunek jest traktowany jako mapa bitowa i nie może być skalowany. Zmiana wielkości obiektu na rysunku jest możliwa jedynie przez skasowanie starego i narysowanie nowego obiektu. Powiązanie graficznych elementów rysunku ze zmiennymi wymaga umieszczenia na nim kolejnego zestawu obiektów odpowiadających konkretnym zmiennym z bazy danych. Dużym minusem jest również brak możliwości konwersji rysunków utworzonych za pomocą innych edytorów, a także brak możliwości zmiany rozdzielczości - zmiana rozdzielczości ekranu komputera, na którym jest uruchomiona aplikacja wizualizacyjna pociąga za sobą konieczność narysowania od nowa wszystkich plansz wizualizacyjnych.

- Touch Links - Value Input <input type="checkbox"/> Discrete <input type="checkbox"/> Analog <input type="checkbox"/> String Value Slider <input type="checkbox"/> Vertical <input type="checkbox"/> Horizontal Touch Pushbutton <input type="checkbox"/> Discrete Value <input type="checkbox"/> Action <input type="checkbox"/> Show Window <input type="checkbox"/> Hide Window	Line Color <input type="checkbox"/> Discrete <input type="checkbox"/> Analog <input type="checkbox"/> Discrete Alarm <input type="checkbox"/> Analog Alarm	Fill Color <input type="checkbox"/> Discrete <input type="checkbox"/> Analog <input type="checkbox"/> Discrete Alarm <input type="checkbox"/> Analog Alarm	Text Color <input type="checkbox"/> Discrete <input type="checkbox"/> Analog <input type="checkbox"/> Discrete Alarm <input type="checkbox"/> Analog Alarm	
	Object Size <input type="checkbox"/> Height <input type="checkbox"/> Width	Location <input type="checkbox"/> Vertical <input type="checkbox"/> Horizontal	Percent Fill <input type="checkbox"/> Vertical <input type="checkbox"/> Horizontal	
	Miscellaneous <input type="checkbox"/> Visibility <input type="checkbox"/> Blink <input type="checkbox"/> Orientation <input type="checkbox"/> Disable	Value Output <input type="checkbox"/> Discrete <input type="checkbox"/> Analog <input type="checkbox"/> String		

Rys. 2. Konfigurowanie zmiennych (program InTouch)
 Fig. 2. Variable configuration (InTouch)

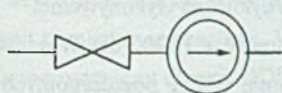
Następna wersja P1200 jest pozbawiona większości wad swojego poprzednika. Na pewno nie bez wpływu na to było przejście do środowiska Windows. Podobnie jak w poprzedniej

wersji, użytkownik ma do dyspozycji zestaw podstawowych elementów możliwych do umieszczenia na rysunku. Ich wygląd jest w pełni konfigurowalny - można zmieniać zarówno położenie i rozmiar obiektów, jak również np. kolor wypełnienia czy grubość linii, którą są one rysowane. Podobnie bez większych trudności można uzależnić wygląd elementu rysunku od wartości zmiennej procesowej. Przewidziano również możliwość grupowania elementów na ekranie i zapisywania takich grup jako symboli, które mogą być wykorzystane w następnych aplikacjach. Mimo zaawansowanych możliwości edytora, jego efektywne używanie wymaga pewnej praktyki i nie jest całkowicie intuicyjne.

Porównywalne możliwości posiada edytor graficzny programu InTouch. Tutaj również twórca wizualizacji umieszcza na ekranie obiekty, których atrybuty są w pełni konfigurowalne. Na rysunku 2 przedstawione zostało okienko dialogowe służące temu celowi.

Ma ono zawsze taką samą postać, chociaż w zależności od rodzaju konfigurowanego obiektu, część opcji jest niedostępna. Praktycznie cały rysunek, łącznie ze szczegółową konfiguracją obiektów, może być wykonany za pomocą myszki, co wpływa pozytywnie na szybkość tworzenia aplikacji. Podobnie jak w przypadku P1200, tutaj również istnieje możliwość grupowania obiektów na rysunku i tworzenia w ten sposób swojego zestawu symboli graficznych.

Jeśli chodzi o program ASIX, to mimo iż pracuje pod kontrolą DOS-u, jego edytor graficzny mocno przypomina edytory w programach dla Windows. Ma on również podobne do



Rys. 3. Symbol pompa-zawór

Fig. 3. Diagram of pump-valve

nich możliwości. Pewną innowacją jest dostarczenie przez producenta serii symboli gotowych do użycia w aplikacjach wizualizacyjnych takich jak stacyjka czy przełącznik. Edytor ma również możliwość definiowania makr, co może przyspieszyć najczęściej wykonywane operacje.

Wszystkie opisane wcześniej edytory dają użytkownikowi możliwość tworzenia własnych symboli, pod warunkiem jednak, że składają się one z najbardziej podstawowych elementów, takich jak linia czy okrąg. Można utworzyć swoje symbole np. pompy i zaworu. Nie jest jednak możliwe utworzenie symbolu zawierającego inne, wcześniej utworzone symbole (np. zestaw pompa-zawór - rys. 3). Taką możliwość będzie miał edytor programu powstającego w firmie PROLOC.

4. Programy użytkownika

W aplikacji wizualizacyjnej czasem istnieje potrzeba przeniesienia części logiki systemu ze sterownika przemysłowego do programu wizualizacyjnego. Dzieje się tak wtedy, gdy aplikacja oprócz suchego przedstawienia danych przychodzących z procesu przemysłowego, powinna wykonywać na nich jeszcze jakieś operacje. Przykładem takiego działania może być uzależnienie stanu pewnego elementu na ekranie od wartości kilku zmiennych procesowych. Fragment programu napisanego przez twórcę aplikacji może analizować wartości tych zmiennych i wpływać na tej podstawie na wygląd elementu na ekranie.

Wszystkie opisywane programy są wyposażone w proste języki programowania, pozwalające dokonywać operacji na zmiennych procesowych.

W przypadku obu wersji P1200 są to odmiany Basic-a (w wersji dla Windows jest to „SCADA Basic”) rozszerzone o dodatkowe funkcje służące do operacji na zmiennych procesowych i do komunikowania się ze sterownikiem przemysłowym. Programy napisane w tych odmianach Basic-a mogą być wykonywane cyklicznie lub po zaistnieniu określonej sytuacji, takiej jak np. wydanie komendy przez operatora systemu.

InTouch oferuje język podobny do Pascala. Programy w nim pisane zwane są tutaj skryptami. Istnieje główny skrypt dla całej aplikacji, a pozostałe są dla niego pewnego rodzaju podprogramami. Tutaj również istnieje możliwość przyporządkowania skryptu jakiemuś zdarzeniu (np. naciśnięcie lub zwolnienie klawisza) lub jego cykliczne wykonywanie.

Pewną ciekawostką stanowią edytory do pisania programów w P1200 dla Windows i w InTouch-u. Oprócz tradycyjnego wpisywania kodu programu, wybór poszczególnych słów kluczowych, a nawet nazw zmiennych, jest możliwy za pomocą myszki.

Program ASIX został wyposażony w język przetwarzania danych procesowych o nazwie ASTEL. Służy on do przetwarzania danych archiwalnych i bieżących. Umożliwia obliczanie wyrażeń arytmetycznych z użyciem danych procesowych. Obliczanie może być przeprowadzane interakcyjnie lub za pomocą napisanego wcześniej programu, wywoływanego na żądanie.

5. Drajwery komunikacyjne

Drajwery komunikacyjne stanowią część aplikacji odpowiedzialną bezpośrednio za transmisję danych między sterownikiem przemysłowym a komputerem wizualizacyjnym. Chodzi

tutaj zarówno o właściwe odwzorowanie obszarów pamięci sterownika w bazie zmiennych procesowych, jak również o przekazanie do sterownika zmian stanów zmiennych wymuszonych np. przez operatora systemu.

Na rynku istnieje bardzo duża liczba protokołów transmisji. Właściwie każdy producent sterowników przemysłowych dostarcza z nimi swój protokół transmisji. Żaden z nich nie zdobył na tyle dominującej pozycji, aby ustanowić w tej dziedzinie szerzej używany standard. Być może stanie się nim zdobywający coraz większą popularność standard sieci FIP. Ze względu na tak dużą różnorodność protokołów transmisji producenci pakietów wizualizacyjnych nie ograniczają się przeważnie do jednego drajwera komunikacyjnego, lecz dają do wyboru kilka standardów.

W przypadku P1200 dla DOS-a użytkownik ma do wyboru trzy rodzaje drajwerów:

- N10,
- NBUS,
- FIP.

Pierwsze dwa umożliwiają połączenie ze sterownikami firmy CEGELEC, ostatni z szeroką gamą urządzeń wyposażonych w interfejs sieci FIP. Są one dostarczane jako programy rezydentne uruchamiane przed właściwym programem wizualizyjnym. Komunikują się z aplikacją poprzez pamięć dzieloną.

Podobnie jest rozwiązana komunikacja w wersji okienkowej programu P1200. Tutaj jednak użytkownik ma do wyboru kilkanaście standardów. Możliwa jest również wymiana informacji pomiędzy aplikacjami. Wykorzystywane są w tym celu windowsowe mechanizmy dynamicznej wymiany danych DDE i NETDDE. Dołączanie drajwerów komunikacyjnych w postaci osobnych modułów umożliwia łatwą ich wymianę, bez konieczności ingerowania w strukturę aplikacji.

Program InTouch komunikuje się ze sprzętem również za pomocą osobno uruchamianych modułów, korzystając, podobnie jak P1200, z dynamicznej wymiany danych DDE.

Z kolei ASIX dostarczany jest z dwoma protokołami pozwalającymi komunikować się ze sterownikami firmy Siemens serii SIMATIC. Dalsze protokoły są w trakcie opracowywania.

Program powstający w firmie PROLOC ma obecnie zaimplementowane również dwa protokoły:

- MODBUS,
- FIP.

Są one obsługiwane przez dynamicznie dołączane biblioteki typu DLL, tak więc ich wymiana jest również bardzo prosta.

6. Archiwizacja i prezentacja danych

Archiwizacja pojawiających się alarmów i zmian stanów zmiennych procesowych, jak również przedstawianie zależności tych zmian od czasu, to cechy wszystkich programów wizualizacyjnych. Poszczególne rozwiązania różnią się od siebie jedynie szczegółami implementacyjnymi oraz bogactwem dodatkowych możliwości oferowanych użytkownikowi.

Jeśli chodzi o wykresy czasowe, zwane również trendami, to dzielą się one na dwa rodzaje:

- bieżące - przedstawiają aktualny przebieg zmiennej w zadanym oknie czasowym, odświeżany z zadaną częstością,
- archiwalne - przedstawiają przebieg zmiennej na podstawie zarchiwizowanych zapisów zmian jej wartości.

Na planszach wizualizacyjnych są również przedstawiane awaryjne stany zmiennych, zwane alarmami. Alarmy są powodowane przez zmianę stanu zmiennych logicznych lub przez przekroczenie określonych limitów przez zmienne analogowe. Użytkownik może przeważnie określać hierarchię pojawiających się alarmów, jak również filtrować je według różnorodnych kryteriów. Teraz kilka słów o poszczególnych rozwiązaniach.

Program P1200 daje w zasadzie jedynie podstawowe możliwości archiwizacji i tworzenia wykresów czasowych. Trendy bieżące mogą być tworzone z sekundową rozdzielczością, natomiast dla trendów archiwalnych minimalny okres pomiędzy zmianami wartości zmiennej wynosi 10 sekund. Maksymalnie 512 zmiennych logicznych i analogowych może brać udział w tworzeniu wykresów.

Alarmy mogą mieć określony jeden z dwóch priorytetów i mogą być przyporządkowane do jednej z trzech grup, zdefiniowanych przez twórcę aplikacji. Na podstawie przynależności do tych grup możliwe jest wybiórcze maskowanie alarmów.

Wersja P1200 dla Windows daje nieco większe możliwości manipulowania danymi. Archiwizowane mogą być zmiany bitów i alarmów, wysyłanie poleceń i tekstu oraz uruchamianie programów użytkownika. Na życzenie czas pojawiania się zdarzeń może być zapamiętywany z milisekundową dokładnością. Istnieje również mechanizm tworzenia trendów bieżących i archiwalnych. Dane mogą być próbkowane z określoną częstotliwością (minimum 1 sekunda) lub zapamiętywane tylko w momentach ich zmian. Dostępna jest szesnastopoziomowa hierarchia alarmów. Oprócz oglądania poszczególnych alarmów na planszach wizualizacyjnych, możliwe jest także obejrzenie całej ich listy w osobnym oknie. Program umożliwia również filtrowanie alarmów według priorytetu, a także ich natury i dziedziny. Ze zarchiwizowanych danych można tworzyć raporty.

Program InTouch ma podobne możliwości archiwizowania i przedstawiania danych, jak uprzednio omówiony P1200 dla Windows. Oferuje on nieco większe możliwości konfigurowania wyglądu listy alarmów. Oprócz tego zmienne przedstawiane na wykresach czasowych mogą być próbkowane z milisekundową rozdzielczością.

W przypadku programu ASIX do archiwizacji używany jest osobny moduł o nazwie ASPAD. Jego konfigurowanie, tzn. określenie nazw archiwizowanych zmiennych, częstotliwości ich próbkowania i czasu przechowywania następuje przez utworzenie plików tekstowych. Dane gromadzone przez ten moduł są używane do tworzenia przebiegów bieżących i archiwalnych. Do graficznej ich prezentacji użytkownik może wybrać wykresy liniowe lub słupkowe.

System alarmów programu ASIX ma między innymi możliwość podziału alarmów na grupy, czasowego filtrowania wybranych alarmów (np. na czas rozruchu systemu) oraz selekcji alarmów według czasu pojawienia się, tekstu, typu, statusu i grupy. Podobnie jak w przypadku poprzednich programów, alarmy mogą być generowane przez zmianę pojedynczego bitu lub przez przekroczenie limitów przez zmienną analogową.

Użytkownik ma również duże możliwości tworzenia raportów ze zgromadzonych danych. Program oferuje do tego celu specjalny język generowania raportów o nazwie ASTER.

7. Podsumowanie

Przedstawiony przegląd możliwości programów wizualizacyjnych nie wyczerpuje na pewno poruszonego tematu. Przegląd zawiera opis kilku bardziej popularnych programów. Nie było moim celem tworzenie rankingu pakietów dostępnych na rynku. Chciałem raczej przedstawić ewolucję, jaką przeszły tego typu programy na przestrzeni ostatnich kilku lat. Jak można zauważyć na przykładzie wybranych pakietów, większość rozwiązań opiera się na możliwościach jednej z najbardziej obecnie popularnych graficznych platform programowych, jaką jest Windows. Wyjątkiem w tym gronie jest pakiet ASIX, działający w DOS-ie. W tej chwili nie można jednoznacznie stwierdzić, które z rozwiązań jest lepsze: czy pozbycie się narzutu programowego Windows przy pewnym ograniczeniu możliwości wymiany informacji, czy też przyjęcie tegoż narzutu, za to z dużym stopniem standaryzacji produktów.

LITERATURA

- [1] Dokumentacja programu P1200, wersja dla DOS.
- [2] Dokumentacja programu P1200, wersja dla Windows.
- [3] Podręcznik użytkownika programu ASIX, ASKOM, Gliwice 1995.
- [4] Dokumentacja programu powstającego w firmie PROLOC,
- [5] Praca zbiorowa: Rozproszone systemy komputerowe, PRO-net, Gliwice 1994.

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 4 stycznia 1996 r.

Abstract

The paper describes main features of SCADA programs. I didn't want to create a ranking of such programs, but I wanted to show the way of development of these programs.

Figure 1 shows schematic structure of visualisation application. Such an application consists of four basic elements:

- graphical view of industrial process,
- real-time database,
- connections between variables and graphical symbols,
- communication with industrial equipment.

In further part of the paper I give a more detail description of ways of these elements creation in some SCADA programs. So the following features are presented:

- creating and structure of real-time database,
- graphical editor,
- writing user programs,
- communication drivers.