STUDIA INFORMATICA 2000
Volume 21 Number 1 (39)

Andrzej KWIECIEŃ
Politechnika Śląska, Instytut Informatyki

PRZEMYSŁOWE SIECIOWE SYSTEMY ROZPROSZONE CZASU RZECZYWISTEGO. CECHY I WYMAGANIA

Streszczenie. Artykuł klasyfikuje i przedstawia wymagania stawiane przemysłowym rozproszonym systemom czasu rzeczywistego. Zwraca uwagę na podstawowe składniki takich systemów czasu rzeczywistego oraz na znaczenie ich prawidłowego zaprojektowania i parametryzacji dla poprawnego działania.

INDUSTRIAL DISTRIBUTED REAL-TIME SYSTEMS. CHARACTERISTICS AND REQUAIREMENTS

Summary. The paper classifies and presents requirements for distributed real-time systems. The basic theirs elements describe also and pay attention to correctly process of designing and parameterisation the real- time systems.

1. Wstęp

Stale wzrastająca niezawodność i moc obliczeniowa systemów komputerowych skłania do szerszego ich stosowania w przemyśle. Systemy komputerowe realizują bardzo odpowiedzialne zadania, a współczesna technologia stawia coraz poważniejsze wymagania związane przede wszystkim z gwarantowanym i nieprzekraczalnym czasem realizacji pojedynczego cyklu sterowania bądź regulacji. To właśnie zmusza do przeprowadzania bardzo szczegółowej analizy możliwości systemu informatycznego. Tym bardziej że owe systemy, z punktu widzenia informatyki, są klasycznymi systemami rozproszonymi czasu rzeczywistego stosującymi sieć komputerową jako podstawowe medium wszelkiego rodzaju wymian informacji.

Pojęcie „system rozproszony czasu rzeczywistego" zawiera w sobie dwa elementy:

- system rozproszony,
- system czasu rzeczywistego.

System rozproszony charakteryzuje się tym, że wiele procesów jest realizowanych na wielu procesorach. Może temu towarzyszyć, aczkolwiek nie musi, rozproszenie terytorialne, które jest tak typowe dla obiektów przemysłowych, zarządzanych i sterowanych za pomocą komputerów. Procesy rozproszone są koordynowane przez grupę interprocesów komunikacyjnych i synchronizacyjnych. Korzyści wynikające ze stosowania systemów rozproszonych są następujące:

- zwiększona moc obliczeniowa wynikająca ze stosowania przetwarzania równoległego,
- zwiększona niezawodność, istnieje bowiem możliwość takiego zaprojektowania systemu, aby w chwili awarii funkcje nie działającego elementu (węzła) systemu zostały przejęte przez inne, działające,
- zwiększona adaptacyjność (rekonfigurowalność) dzięki możliwościom modyfikacji programów i łatwemu podziałowi zasobów.

W pełni rozproszonym systemem nazywamy taki, w którym sterowanie, sprzęt i dane są rozproszone. Istnieją bowiem systemy, które nie w pełni są systemami rozproszonymi. Do nich zaliczyć należy [1]:

- systemy typu SIMP - w których sprzęt jest zmultiplikowany, ale występuje scentralizowane sterowanie,
- systemy typu MIMD - w których sprzęt jest scentralizowany, a rozproszone jest sterowanie.

Systemy rozproszone można podzielić również na:

- heterogeniczne lub homogeniczne,
- scentralizowane (np. MASTER-SLAVE, KLIENT-SERVER),
- zdecentralizowane, w których węzły systemu są autonomiczne.

Idąc dalej, można dokonywać coraz to innych klasyfikacji, ale wspólną cechą lub tendencją obowiązującą przy projektowaniu tych systemów - to dążność do stosowania sieci komputerowej w celu tworzenia grup stacji roboczych i komunikacji międzyprocesowej.

System czasu rzeczywistego charakteryzuje się silnymi ograniczeniami czasowymi (ang. „timing constraints"). Czas jako parametr staje się w tych systemach krytyczny. Typowe systemy czasu rzeczywistego zawierają podsystemy kontrolująco-sterujące (np. specjalizowane kontrolery komputerowe), a także podsystemy kontrolowane i sterowane (np. warstwa fizyczna obiektu).

Obie grupy podsystemów podlegają silnej interakcji, którą można opisać trzema operacjami:

- zbieranie danych z warstwy fizycznej obiektu (ang. „*sampling*"),
- natychmiastowe przetwarzanie zebranych danych i ewentualne uruchamianie procesów obliczeniowych (algorytmów) (ang. „*processing*"),
- odpowiedź systemu skierowana do obiektu warstwy fizycznej będąca wynikiem działania warstwy obliczeniowej (ang. „*responding*").

Wymienione trzy operacje, co również charakteryzuje system czasu rzeczywistego, muszą zakończyć się w określonym, wyspecyfikowanym czasie.

Systemy czasu rzeczywistego można podzielić na dwie grupy:

- systemy, w których istnieje znaczna tolerancja łamania ograniczeń czasowych (ang. „*soft real-time systems*"),
- systemy, w których tolerancja łamania ograniczeń czasowych albo nie jest dopuszczalna, albo niewielka (ang. „*hard real-time systems*").

W systemach ze znaczną tolerancją łamania ograniczeń czasowych ich przekroczenie nie pociąga za sobą znacznych konsekwencji. System pracuje poprawnie nawet przy przekroczeniu ograniczeń, ale nie w sposób drastyczny.

Typowymi przykładami takich systemów są:

- system zdalnego dostępu do danych,
- system rezerwacji biletów,
- system bankomatów.

W systemach bez tolerancji łamania ograniczeń ich przekroczenie grozi znacznymi konsekwencjami.

Typowymi takimi systemami są:

- systemy, od poprawności działania których zależy np. ludzkie życie,
- systemy kontroli ruchu lotniczego,
- systemy sterowania i regulacji większością procesów przemysłowych,
- systemy monitorowania pacjentów,
- systemy z krytycznym czasem reakcji,
- systemy robotów,
- systemy telefonii.

Jak już wspomniano, w systemach czasu rzeczywistego pojęcie czasu jest niezwykle ważne. Występuje on w czterech aspektach:

- dostęp do zegara,
- opóźnienia przetwarzania,
- detekcja opóźnień („time-out-ów"),
- specyfikacja ograniczeń czasowych i określenie czasu reakcji.

Dostęp do zegara może być absolutny przez dostęp do zegara czasu rzeczywistego albo pośredni przez pomiar, lub detekcję czasu wykonywania pętli. Dostęp może być również związany z synchronizacją zegarów w węzłach systemu. Jest to niezwykle trudny i typowy problem przemysłowych systemów rozproszonych czasu rzeczywistego, gdzie musi być brany pod uwagę tak zwany „najgorszy przypadek" [2] i wszelkie operacje muszą być odniesione do zegara „najwolniejszego".

Zdolność opóźniania przetwarzania jest związana z zagadnieniem kolejkowania. Metoda kolejkowania eliminuje bowiem tak zwane stany „zbędnego oczekiwania"(ang. *"busy wait"*). Proces może skierować zadanie do kolejki, które będzie wykonane później już jako zdarzenie zamiast generować sztuczne opóźnienie w postaci stanów *„busy wait"*.

Detekcja tak zwanych *„time-out"*-ów dotyczy zdolności procesu do rejestracji braku pojawienia się zdarzenia i w konsekwencji do „znieczulania" się na pojawienie się określonego zdarzenia.

Specyfikacja ograniczeń czasowych jest związana z respektowaniem ograniczeń czasowych, narzucanych przez warstwę fizyczną obiektu. Jest to związane z minimalnym i maksymalnym czasem realizacji pętli programu (ang. *"deadline specification"*).

2. Podstawowe problemy projektowania rozproszonych systemów czasu rzeczywistego

Podstawowe problemy projektowania i późniejszej realizacji systemów czasu rzeczywistego związane są z ich podstawowymi właściwościami, do których możemy zaliczyć:

- działanie w sposób ciągły w bardzo trudnych warunkach,
- ostre ograniczenia czasowe,
- interakcję pomiędzy procesami asynchronicznymi,
- przewidywalność opóźnień i warunki tak zwanego „wyścigu",
- nieokreśloność i brak powtarzalności (cykliczności) zachowań,
- globalność zegara i stanu obiektu,
- wielowątkowość interakcji procesów.

Większość rozproszonych systemów czasu rzeczywistego, a w tym również systemów sterowania i monitorowania procesów przemysłowych, musi pracować ciągle i zachowywać interakcję ze sterowanym obiektem. Mówiąc o trudnych warunkach, należy mieć na myśli przede wszystkim szereg wzajemnych czasowych zależności pomiędzy interprocesami. Zależności te objawiają się w sposób wyjątkowo niebezpieczny i „dokuczliwy" w chwilach stanów nieustalonych obiektu, co powoduje szereg wywołań zadań, które są ze sobą powiązane kolejnością i czasem ich realizacji. Ma to na przykład miejsce w chwilach pojawiania się tak

zwanej „obiektowej lawiny zdarzeń". Trudne warunki pracy dotyczą również, co ma miejsce w zastosowaniach przemysłowych, warunków środowiskowych i klimatycznych, w których sprzęt informatyczny ma pracować.

Ostre ograniczenia czasowe wpływają na „dokładność" i poprawność wykonywania zadań przez rozproszone systemy czasu rzeczywistego. „Dokładność" i poprawność pracy jest określona nie tylko przez liczbę i szybkość procesorów sterujących podwarstwami całego systemu, ale także przez ograniczenia czasowe narzucone przez czas obsługi obiektu (otoczenia).

Rozproszone systemy czasu rzeczywistego są projektowane do interakcji z warstwą fizyczną, a procesory tych systemów są rozproszone terytorialnie. Procesy asynchroniczne, zawierające wewnętrzne systemy rozproszone czasu rzeczywistego i zewnętrzne procesy fizyczne, komunikują się pomiędzy sobą poprzez wymianę komunikatów. Sekwencje zdarzeń w procesach asynchronicznych są trudne do przewidzenia i często przyjęte założenia na etapie projektowania są naruszane w czasie działania.

Z powodu trudnego do przewidzenia natężenia ruchu w sieci komunikacyjnej i dystansu pomiędzy dwoma węzłami komunikacyjnymi systemu, opóźnienia wnoszone przez komunikację międzyprocesorową są nieprzewidywalne, ale nie można ich pominąć. Stąd konieczność stosowania do analizy „metody najgorszego przypadku" [2]. Dodatkowo może wystąpić zjawisko „wyścigu", gdy dwa procesy dzielą się tymi samymi zasobami. Tak więc sekwencje zdarzeń specyfikowane na etapie projektowania, a służące synchronizacji podprocesów, mogą się zmieniać podczas normalnej pracy systemu.

Z powodu trudności przewidywania opóźnień komunikacyjnych i warunków „wyścigu" pomiędzy procesami i procesorami zachowanie systemów rozproszonych czasu rzeczywistego (rezultaty wykonania) może być nieokreślone. Może być bowiem tak, że ponowne wykonanie tego samego programu, z tymi samymi stanami wejść, niekoniecznie musi generować te same rezultaty.

Każdy procesor systemu ma własny zegar działający niezależnie od zegarów innych procesorów. Jest niezwykle trudno określić precyzyjnie czas globalny, który jest niezbędny dla określenia właściwych stanów globalnych i w procesie diagnozowania i monitorowania pracy całego systemu.

W przeciwieństwie do programów sekwencyjnych, mających pojedynczy mechanizm sterowania, systemy rozproszone czasu rzeczywistego cechują wielokrotnie przeplatające się sekwencje (przepływy) sterowania. Także strumienie sterowania są swobodnie lub rygorystycznie powiązane z protokołami komunikacyjnymi (przez pamięć lub za pomocą komunikatów). Dodatkowo istnienie wielowątkowości jest utrudnieniem przy analizie i monitorowaniu pracy systemu.

Podsumowując, systemy rozproszone czasu rzeczywistego odróżniają się zasadniczo od innych systemów. Podstawowymi wyróżnikami są ograniczenia czasowe. Najbardziej istotne zagadnienie, niezwykle ważne przy analizie i śledzeniu pracy systemu, to znalezienie miejsc jego pracy, gdzie są pogwałcane ograniczenia czasowe. Komunikacja międzyprocesorowa i międzyprocesowa to najważniejsze źródła pogwałceń zależności czasowych.

3. Przemysłowe rozproszone systemy czasu rzeczywistego

Jak powszechnie wiadomo, podstawowym elementem przemysłowego systemu sterowania jest komputer przemysłowy lub jego odmiana, jaką jest sterownik swobodnie programowalny (PLC), albo specjalizowane urządzenie mikroprocesorowe wyposażone w koprocesor komunikacyjny. Rozproszenie systemu uzyskuje się w sposób naturalny, bo o tym decyduje obiekt przemysłowy, w którym da się wyodrębnić terytorialnie oddzielone procesy technologiczne, które będą sterowane niezależnie i w dużym stopniu autonomicznie przez węzły systemu komputerowego. Węzłem systemu komputerowego jest zazwyczaj pojedynczy sterownik (PLC) lub ich grupa. Dlatego cały system staje się natychmiast rozproszonym systemem w sensie informatycznym, gdyż następuje tutaj rozproszenie zarówno procesorów, jak i zasobów i sterowania. Oprócz komputerów przemysłowych (sterowników PLC) występują w takim systemie stacje robocze lub inżynierskie, których zadaniem jest tak wizualizacja procesu przemysłowego, monitorowanie jego pracy, archiwizacja zdarzeń i alarmów, jak i generowanie rozkazów kierowanych do obiektu przemysłowego. Wymagania technologii idą w kierunku nie tylko precyzyjnego, w sensie jakościowym, sterowania związanego z poprawnym realizowaniem algorytmów numerycznych, ale również w kierunku narzucania bardzo ostrych ograniczeń czasowych. A więc nie chodzi tylko o to, aby zostały poprawnie wyliczone nowe wartości wektora stanu obiektu, ale i o to, aby stało się to w nieprzekraczalnym, zadanym interwale czasu. Jawi się nam zatem już nie tylko przemysłowy rozproszony system sterowania, ale również rozproszony system sterowania czasu rzeczywistego.

Dokonując analizy pracy takiego systemu, która ma służyć między innymi prawidłowemu jego zaprojektowaniu, należy dokonać podziału na elementarne składniki, do których można zaliczyć:

- węzeł systemu, w skład którego wchodzi komputer przemysłowy (PLC),
- system komunikacyjny, którego elementem jest protokół komunikacyjny,
- stacja operatorska z systemem wizualizacyjnym zbudowanym na bazie systemu operacyjnego czasu rzeczywistego.

Każdy z wymienionych elementów zachowuje znaczne możliwości pracy autonomicznej, ale stanowią one składniki systemu rozproszonego i muszą podlegać tym samym wymaganiom, o których była mowa w poprzednich rozdziałach. Projektując przemysłowy, rozproszony system czasu rzeczywistego, należy każdy z wymienionych elementów poddać niezależnej analizie, zmierzającej do określenia granicznego czasu T_G reakcji całego systemu, a tym samym do określenia granic jego stosowalności.

4. Węzeł systemu

Jak już wspomniano, węzłem systemu jest sterownik przemysłowy (PLC) lub grupa sterowników. Rolą sterownika jest realizacja programu rezydującego w jego pamięci operacyjnej. Program jest zakodowanym, numerycznym algorytmem sterowania. Programowanie sterownika jest niczym innym, jak tworzeniem małego systemu operacyjnego czasu rzeczywistego, którego podstawową cechą jest cykliczność realizacji. Tworząc oprogramowanie, należy kierować się tymi samymi kryteriami co przy tworzeniu dużych systemów operacyjnych czasu rzeczywistego. Szczególną uwagę należy zwracać na czas realizacji podstawowej pętli programu.[3]. Czas trwania pętli zależy wprost proporcjonalnie od czasu realizacji aplikacji. To znów jest zależne od złożoności algorytmów, które mają być aplikowane i od umiejętności programisty, którego rolę trudno przecenić. Doskonała znajomość parametrów sterownika, jego architektury pozwala tworzyć oprogramowanie w sposób optymalny, to znaczy minimalizować czas realizacji programu przy jednoczesnej minimalizacji zajętości pamięci. Czas cyklu realizacji programu ma decydujący wpływ na sprawną komunikację. Sterownik, a tym samym węzeł systemu komunikuje się z innymi węzłami za pośrednictwem koprocesora sieci. Koprocesor natomiast ma dostęp do zasobów macierzystej jednostki centralnej tylko raz na jeden cykl realizacji pętli programu. Zatem, im dłuższy jest cykl pojedynczej realizacji programu systemowego (aplikacja jest systemem operacyjnym czasu rzeczywistego), tym rzadziej koprocesor może kontaktować się z zasobami i tym później jest w stanie wyemitować niezbędne dane.

Na granicy koprocesor-jednostka centralna węzła systemu zachodzi cały szereg zjawisk, których poznanie jest niezbędne do przeprowadzenia prawidłowej analizy pracy systemu. Zjawiska te, jak choćby przetwarzanie ramki czy generacja sumy kontrolnej, nie tylko należy znać, ale również należy precyzyjnie określić ich czas trwania. Wpływa to, jak łatwo się domyślić, w sposób znaczący na sprawność systemu komunikacyjnego

5. Protokół komunikacyjny

Protokół komunikacyjny jest kolejnym elementem rozproszonego systemu czasu rzeczywistego. Aby spełnić warunki systemu czasu rzeczywistego, musi to być protokół o zdeterminowanym w czasie dostępie. Należy przez to rozumieć, że protokół gwarantuje każdemu uczestnikowi procesu wymiany informacji możliwość emisji i odbioru danych w nieprzekraczalnym, gwarantowanym i znanym czasie. Nie oznacza to bynajmniej, że parametr ten jest cechą protokołu. To dopiero analiza przepływu danych w systemie da odpowiedź, ile ten czas wynosi, a protokół zapewni jedynie, że nie zostanie on przekroczony. To, że sieć jest w stanie przesłać określoną ilość informacji w pewnym czasie, którego wielkość zależy choćby od parametrów transmisji, takich jak prędkość czy długość ramki danych, nie determinuje sprawności komunikacji. Nie należy również rozpatrywać protokołu komunikacyjnego w oderwaniu od koprocesora sieci. Tu podobnie jak w węźle systemu zachodzi szereg zjawisk, które będą wpływać na czas wymiany informacji. Ważne jest, aby znać budowę koprocesora i jego wszystkie parametry czasowe i pewne cechy konstrukcyjne, jak choćby liczbę buforów do transmisji i odbioru.

Współcześnie istnieje kilka powszechnie znanych i akceptowanych protokołów o zdeterminowanym w czasie dostępie do medium. Do nich należy zaliczyć protokoły z „krążącym żetonem", typu „Master-Slave" czy „Producent-Dystrybutor-Konsument". Na marginesie tej informacji należy pamiętać, że na wybór protokołu będzie miał wpływ nie tylko determinizm czasowy, ale również rodzaj aplikacji. Będą bowiem rozwiązania „szybsze" jak i „wolniejsze", do których należy dobrać odpowiednią sieć, będącą najtańszym rozwiązaniem, ale spełniającym wszelkie wymagania czasowe.

6. Stacja operatorska

Stację operatorską należy rozpatrywać w dwóch kontekstach. Pierwszym z nich jest pojęcie stacji jako węzła systemu. Przy takim podejściu pozostają w mocy wszystkie uwagi zamieszczone w poprzednich dwóch rozdziałach. Wyodrębniając w niej moduł czy podproces komunikacji, należy dokonać analizy czasowej związanej z procesem odbioru jak i nadawania. Można i trzeba taki podproces potraktować jako koprocesor sieci i wniknąć w jego budowę, aby znaleźć odpowiedź na mniej więcej takie same pytania, jakie powstają przy analizowaniu klasycznego koprocesora. A więc należy obliczyć czasy trwania takich procesów, jak:

- czas detekcji ramki,
- czas analizy odebranej ramki danych,

- czas przetwarzania ramki przez odpowiednią warstwę oprogramowania,
- czas dostępu do warstwy aplikacji,
- czas obliczania cyklicznej sumy kontrolnej,
- czas przygotowania ramki do transmisji uwzględniający czas oczekiwania na dostęp do pamięci macierzystej jednostki centralnej,
- czas zapełnienia buforów nadawczych.

Drugim zagadnieniem czy kontekstem, pod kątem którego należy rozpatrywać stację operatorską, to stacja jako jednoprocesorowy system czasu rzeczywistego będący aplikacją zrealizowaną na bazie systemu operacyjnego. Przy tworzeniu takiej aplikacji napotykamy na szereg tych samych problemów związanych z poprawnym projektowaniem systemów czasu rzeczywistego, o których była już mowa. Stacja operatorska charakteryzuje się tym, że składa się z szeregu procesów, które wzajemnie muszą współgrać, mając dostęp do tych samych zasobów, komunikować się między sobą za pomocą pamięci, wykluczać się i blokować w przypadkach konfliktów, przekazywać sterowanie pomiędzy sobą, a wszystko to musi spełniać wymagania systemu czasu rzeczywistego. Nie bez znaczenia jest również współpraca z platformą systemu operacyjnego[4].

Niezwykle ważnym problemem jest tworzenie konkretnej aplikacji użytkowej stacji wizualizacyjnej, która, jak pamiętamy, jest systemem czasu rzeczywistego. Pojawiają się zatem problemy zsynchronizowania aplikacji z resztą systemu rozproszonego, właściwego obciążania systemu wizualizacyjnego, właściwej koincydencji wywoływania specjalizowanych procedur (ang. *„users programs"*), dopuszczalnej liczby obiektów graficznych i ich ewentualnego odświeżania i animowania. Wszystko to razem powoduje, że zaprojektowanie aplikacji wizualizacyjnej nie jest banalne i wymaga wnikliwej analizy i znacznej wiedzy informatycznej.

7. Zakończenie

Niestety, zastosowanie systemów informatycznych w przemyśle do sterowania i regulacji, nie pozwala popaść projektantom w rutynę, gdyż to przede wszystkim rodzaj procesu technologicznego narzuca sposób tworzenia oprogramowania rezydującego w węzłach systemu.

Słowo „niestety" oznacza mniej więcej tyle, że nie ma dwóch absolutnie identycznych procesów technologicznych i nie da się stworzyć ani identycznego oprogramowania sterującego procesem, ani identycznego oprogramowania stacji operatorskiej. Za każdym razem proces tworzenia systemu musi rozpocząć się jego analizą czasową. Nie można zatem stworzyć sztywnych metod projektowania, które miałyby zastosowania w każdym przypadku.

Można jedynie zbudować ogólny model węzła systemu, przedstawić i starać się opisać w sposób formalny wszystkie zjawiska w nim zachodzące, a szczególnie te na styku „aplikacja-koprocesor-protokół komunikacyjny", tak aby proces projektowania sformalizować i przygotować do należytego testowania [5]. Testowanie systemu nie może natomiast mieć miejsca na obiekcie przemysłowym, gdyż, z praktycznego punktu widzenia, jest to nierealne. Tym bardziej należy przywiązywać duże znaczenie do analizy przepływu informacji w systemie, na etapie projektowania.

Od wielu lat są prowadzone w Instytucie Informatyki Politechniki Śląskiej badania nad zagadnieniami, o których jest mowa w niniejszej pracy. Badania dotyczą analizy przepływu informacji w komputerowych, przemysłowych systemach rozproszonych. Rezultaty tych prac to przede wszystkim liczne publikacje, referaty i wystąpienia na seminariach i konferencjach poświęconych tej problematyce. Rezultatem jest zarówno książka [5] na temat analizy przepływu informacji w przemysłowych, komputerowych systemach rozproszonych, jak i praca doktorska [4] dotycząca budowy systemów wizualizacyjnych. Nie bez znaczenia są również praktyczne aplikacje rezultatów tych badań. Zespół pracowników Instytutu Informatyki Politechniki Śląskiej ma na swym koncie kilkadziesiąt przemysłowych aplikacji rozproszonych systemów rozproszonych czasu rzeczywistego, a wśród nich tak odpowiedzialne, jak:

- system sterowania pracą stacji uzdatniania wody w Elektrowni Trzebovice w Republice Czeskiej. System składa się z dziesięciu węzłów połączonych siecią przemysłową, której magistrala ma długość 1200m. W systemie istnieje 2700 zmiennych systemowych przy 4000 zmiennych obiektowych (1995 r.),
- system sterowania i regulacji procesami technologicznymi w EC I- Garbary-Poznań. System składa się z siedmiu węzłów połączonych dwiema wielomagistralowymi sieciami (MODBUS i FIP), w których istnieje 1500 zmiennych systemowych przy 2200 zmiennych obiektowych (listopad 1999 r.).

LITERATURA

1. Smith Y. T.: Distributed Real-Time Systems. John Wiley &Sons New York 1996.
2. Kwiecień A., Bigewski Z., Mrówka Z.: Analiza czasu najgorszego przypadku w sieciach przemysłowych, ZN Pol Śl. s. Informatyka, z. 36, Gliwice 1999.
3. Gaj P.: Szybka sieć przemysłowa a system wizualizacji-problem interfejsu. ZN Pol. Śl s. Informatyka, z. 36, Gliwice 1999.
4. Cupek R.: Metody wizualizacji rozproszonych procesów przemysłowych. Praca doktorska, Instytut Informatyki Politechniki Śląskiej, Gliwice 1998.

5. Kwiecień A.: Analiza przepływu informacji w komputerowych sieciach przemysłowych. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego. Gliwice 1999.

Recenzent: Dr inż. Włodzimerz Boroń

Wpłynęło do Redakcji 10 kwietnia 2000 r.

Abstract

The paper describes the basis problems of industrial distributed real-time systems. As it is well known, the basic component of an industrial control system is an industrial computer or a PLC being its variety or a specific purpose microprocessor unit equipped with a communication co-processor. The system distribution is obtained naturally, since it is decided by the industrial object in which territorially separated technological processes can be distinguished to be controlled independently and to a great extent autonomously by the computer system nodes. The computer system node is generally a single PLC or a group of PLCs. Therefore, the system becomes itself a distributed system from the information technology point of view as both the processors and the control resources are subject to distribution there. Apart from the industrial computers/PLCs there are also in such system working/engineering stations whose role is to visualise the industrial process, monitor its functioning, make history of events and alarms and generate commands toward the industrial object. The technology requirements tend to have not only precise (in terms of quality) control related to a proper realisation of numerical algorithms but also towards establishing very severe time constraints. Thus, the problem is not related to a regular computation of the new object status vector values only but requires also accomplishing such computation within a preset deadline. A distributed real-time control system emerges not only a distributed one.

On making analysis of such a system to support its proper design one should decompose the system into elementary components that will include: the system node embracing the industrial computer (PLC); the communication system with its communication protocol; the operator workstation with the visualisation system built around a real-time operating system.

Every above mentioned component conserves considerable possibilities of autonomous functioning but these components are part of the distributed system and must be subject to the same constraints as described in previous sections. When designing an industrial distributed real-time system one should make an independent analysis of each component to determine the time-out T_G for the system's response and thereby determine the application limit of the system.