ZESZYTY NAUKOWE POLITECHNIKI SLASKIEJ 1996
Seria: AUTOMATYKA z. 118 Nr kol. 1338

Lech JAMROŻ, Jerzy RASZKA
Politechnika Krakowska

METODA SYMULACYJNA OCENY WYDAJNOŚCI CYKLICZNYCH PROCESÓW DYSKRETNYCH

Streszczenie. Artykuł dotyczy problematyki planowania i sterowania produkcją w dyskretnym systemie produkcyjnym w warunkach ograniczonych zasobów. Przedmiotem rozważań jest system typu job-shop o cyklicznie przebiegających procesach dyskretnych. Na gruncie teorii czasowych sieci Petriego opisano występujące w systemie współbieżnie procesy, do synchronizacji których zastosowano regułę wzajemnego wykluczania. Za pomocą symulacji komputerowej przeprowadzono ocenę wydajności pracy rozważanego systemu.

SIMULATION METHOD FOR THE PERFORMANCE EVALUATION OF SYSTEM OF DISCRETE CYCLIC PROCESSES

Summary. The paper deals with the problem of real-time scheduling and control in discrete production system under limited resources. A job-shop system of cyclic discrete processes is considered. A timed Petri nets are used for specification of concurrent cyclic processes with mutual exclusion. Application of computer simulation to the performance evaluation of considered system is proposed.

1. Wstęp

Optymalne wykorzystanie zasobów oraz rozwiazywanie konfliktów zasobowych są podstawowymi zagadnieniami w automatycznym sterowaniu produkcją w dyskretnych systemach produkcyjnych (DSP) [1, 2, 6]. Przyjęcie jednolitego modelu do opisu tych zagadnień pozwoliło na jednoczesne ich rozwiązanie. Rozważany DSP typu job-shop jest przykładem sieciowego systemu działań, w którym operacje powiązane są między sobą ustalonymi uwarunkowaniami czasowymi [4, 5]. W systemach takich znaczący wpływ na ich wydajność ma koordynacja współbieżnie przebiegających procesów. Do zapewnienia bezblokadowego przebiegu tych procesów zastosowano regułę „wzajemnego wykluczania" jako mechanizmu synchronizacji. W celu skonstruowania modelu formalnego DSP i przeprowadzenia symulacji do oceny jego wydajności wykorzystana została teoria sieci

Petriego [8]. Znakowany graf synchronizacji będący podklasą czasowych sieci Petriego, stanowi adekwatną reprezentację przebiegu rozważanej klasy procesów. Jako miarę oceny wydajności DSP przyjęto współczynnik przepustowości zadań w systemie. W terminach czasowych sieci Petriego największą wartość tego współczynnika uzyskuje się dla minimalnego czasu cyklu [7]. Wobec wykładniczej złożoności obliczeniowej ogólnego problemu czasu cyklu zastosowano podejście oparte na symulacji komputerowej [3, 9].

2. Reprezentacja sieciowa procesów. Pojęcia i założenia

2.1. Procesy dyskretne w systemie job-shop

Rozważane są procesy przebiegające w idealizowanym systemie typu job-shop. Zakładamy, że każdy proces dyskretny jest sekwencyjny i cykliczny. Metodą symulacji komputerowej oceniono wydajność pracy takiego systemu. Rozważany system produkcyjny obejmuje zbiór różnych typów zadań $J=\{J_1,J_2,...J_k,...,J_n\}$ oraz zbior różnych typów maszyn $m=\{m_1,m_2,...m_u,...m_r\}$. Proces produkcyjny każdego zadania $J_k \in J$ określony jest przez uporządkowany ciąg operacji $\langle J_k \rangle = \langle m_1(k),m_2(k),...m_i(k),... ,m_r(k)\rangle$, $k=1,2,...,n$, gdzie $m_i(k)$ jest operacją zadania $J_k \in J$ wykonywaną na maszynie $m_i \in m$. Ustalona kolejność zadań na danej maszynie określona jest przez sekwencję $\langle m_u \rangle = \langle ... J_{k-1}(u),J_k(u),J_{k+1}(u)... \rangle$, $u=1,2,...,r$. Czas wykonywania na maszynie m_u operacji zadania J_k wynosi τ_u^k $u=1,2,...r$, $k=1,2,...n$. Zakładamy, że zadania mogą być wykonywane w trybie cyklicznym. Na rys. 1 przedstawiona jest struktura (s+1)- operacyjnego zadania $\langle J_k \rangle = \langle m_0(k),..,m_{i-1}(k),m_i(k),m_{i+1}(k),..,m_s(k)\rangle$. Pojęcie procesu odnosi się do sekwencji czynności (operacji), na które składa się wykonanie zadania $J_k \in J$, spełniającej nałożone warunki technologiczne. Rzeczywisty system DSP cechuje ograniczona zarówno ilość środków zasobowych, jak i liczba procesów. Współbieżna realizacja procesów w warunkach ograniczonych zasobów wymaga określenia reguły synchronizacji zapewniającej bezblokadowy ich przebieg. Do budowy modelu symulacyjnego współbieżnych procesów cyklicznych zostały wykorzystane elementy teorii sieci Petriego.

2.2. Czasowa sieć Petriego

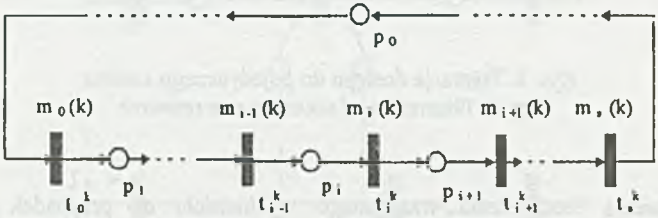
Dla systemów sterujących produkcją [1] czy systemów komputerowych [7] czasowe sieci Petriego są adekwatnym modelem przebiegających w nich dyskretnych procesów. Uwzględniają bowiem równoległość procesów, mechanizmy ich synchronizacji oraz rozdział zasobów. Ponadto pozwalają ocenić wydajność tych systemów.

Sieć Petriego (*SP*) jest uporządkowaną czwórką N=(P,T,F,W), gdzie: $P=\{p_1,p_2,...,p_r\}$ - zbiór miejsc, $T=\{t_1,t_2,...,t_n\}$ - zbiór przejść, $F\subseteq(PxT)U(TxP)$ - zbiór łuków (relacja przepływu) $W:F\rightarrow\{1,2,...\}$ - krotość łuku (funkcja wagowa, przyjmujemy W(f)=1, $f\in F$). Znakowaniem (markowaniem) sieci Petriego jest funkcja $M:p\rightarrow\{0,1,2,...\}$, która przypisuje miejscom nieujemną liczbę znaczników. Funkcja M przedstawiana jest w postaci wektora |p| elementowego, którego i-ta składowa $M(p_i)$ określa liczbę znaczników w miejscu $p_i\in P$. Zakładamy, że każdemu przejściu $t\in T$ przypisana jest zmienna τ_t mierzona w jednostkach czasu. Czasowa znakowana sieć Petriego *TMSP* jest trójką *TMPN*=(N, M_0, T), gdzie: M_0 - znakowanie początkowe, $T:T\rightarrow R^+$ funkcja przypisująca wartość rzeczywistą i dodatnią do każdego przejścia $t\in T$. Przyjmujemy, że reprezentacją graficzną miejsc są okręgi, natomiast przejść prostokąty.

2.3. Znakowany graf synchronizacji

Graf synchronizacji (*GS*) jest to *SP*, dla której N spełnia następującą zależność $|\bullet p|=|p\bullet|=1$, dla każdego $p\in P$, gdzie: $\bullet p=\{t:(t,p)\in F\}$, $p\bullet=\{t:(p,t)\in F\}$. Znakowany graf synchronizacji (*ZGS*) jest podklasą *TMSP*, dla której N jest *GS*.

Symulacja zachowania się rozważanego systemu procesów została oparta na strukturze *ZGS*.

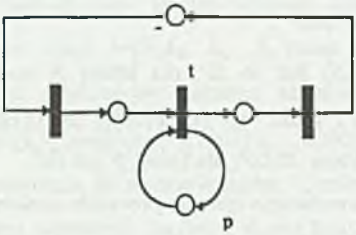


Rys. 1. Model zadania $J_k\in J$ w postaci *ZGS*
Fig. 1. *ZGS* model of job $J_k\in J$

Przyjmujemy, że czas modelowany jest liczbą $\tau_t=T(t)$ i jest przypisany tranzycjom. Wobec tego w *ZGS* na rys. 1. tranzycja t_i^k reprezentuje operację zadania $J_k\in J$ wykonywaną na maszynie $m_i\in m$ w czasie $\tau_i^k=T(t_i^k)$. Miejsca p_i, i=1,2,...,s oraz p_0 reprezentują odpowiednio bufor maszyny $m_i\in m$ oraz środek transportu. Zakładamy również, że zasoby są przydzielane na czas jednej operacji po zakończeniu której są ponownie udostępniane. Tryb dostępu zasobu reprezentowanego przez miejsce p do operacji reprezentowanej przez przejście t przedstawia rys. 2.

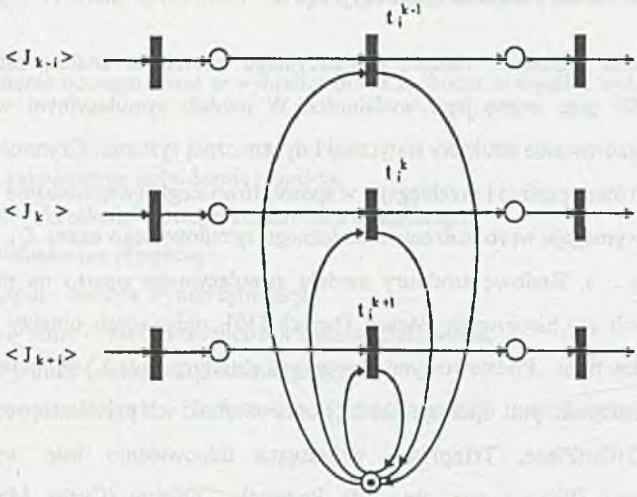
3. Synchronizacja procesów

Procesy komunikują się między sobą w trybie wzajemnego wykluczania. Efektywność takiego mechanizmu komunikowania się dla systemów komputerowych została przedstawiona w pracy [7]. Przyjęty mechanizm synchronizacji wymaga wprowadzenia do sieci dodatkowego miejsca dla każdej jednostki zasobu. Zakładamy, że do realizacji operacji wyrażonej przejściem $t \in T$ niezbędny jest zasób, z którym związane jest miejsce $p \in P$. Istniejący w tym miejscu znacznik określa dostępność danego zasobu. Zasoby przydzielane są tylko na czas określonej operacji. Ponowne wykorzystanie zasobu wiąże się z kolejnym jego przydziałem. Wobec tego w *ZGS* na rys. 2. jeżeli z miejsca p przypisanego zasobowi skierowany jest łuk do przejścia t, to musi być skierowany w stronę odwrotną.

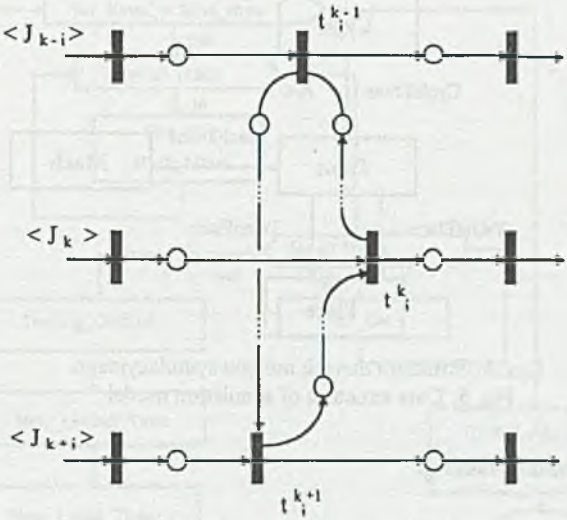


Rys. 2. Ilustracja dostępu do pojedynczego zasobu
Fig. 2. Illustration of access to one resource

Uogólnienie mechanizmu wzajemnego wykluczania na przypadek użytkowania jednego zasobu przez kilka procesów przedstawia rys. 3. Przekazanie sterowania procesom (por. rys. 4.) osiągającym znakowanie, które spełnia warunki dostępności do zasobu, odbywa się zgodnie z regułą wzajemnego wykluczania.



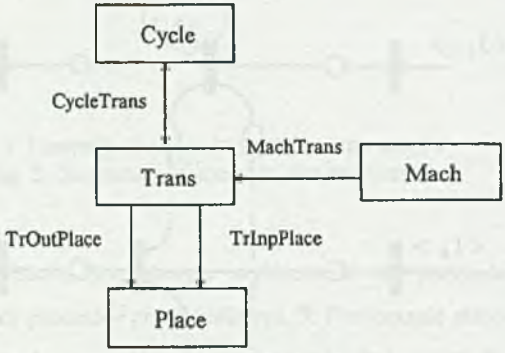
Rys. 3. Ilustracja dostępu do jednego zasobu przez kilka procesów
Fig. 3.Illustration of access to one resource by several processes



Rys. 4. Ilustracja reguły wzajemnego wykluczania
Fig. 4. Illustration of the use of mutual exclusion rule

4. Założenia strukturalne modelu symulacyjnego

Opracowana struktura modelu symulacyjnego umożliwia analizę zachowania się rozważanego DSP oraz ocenę jego wydajności. W modelu symulacyjnym wyodrębniono programowe odwzorowanie struktury statycznej i dynamicznej systemu. Czynności (operacje), które występują równocześnie i przebiegają w sposób równoległy (współbieżnie w skali czasu rzeczywistego), wymagają wprowadzenia niezależnego symulowanego czasu $\zeta_1, \zeta_2, \ldots \zeta_i, \zeta_{i+1}, \ldots$ ($\zeta_i \leq \zeta_{i+1}$). Budowę struktury modelu symulacyjnego oparto na mechanizmach implementowanych w Sieciowych Bazach Danych [10], opisujących obiekty oraz związki zachodzące między nimi. Podstawowymi elementami struktury (rys. 5.) są przejścia (Trans - reprezentujące zdarzenia, por. operacje pkt 2.1) oraz warunki ich palenia się reprezentowane przez relacje TrOutPlace, TrInpPlace wskazujące odpowiednio listę wyjściowych i wejściowych miejsc (Place - por. pkt. 2.2). Pozostałe obiekty (Cycle, Mach) i relacje (CycleTrans, MachTrans) definiują obwody synchronizacji i wykonywane na maszynach zadania.



Rys. 5. Struktura danych modelu symulacyjnego
Fig. 5. Data structure of simulation model

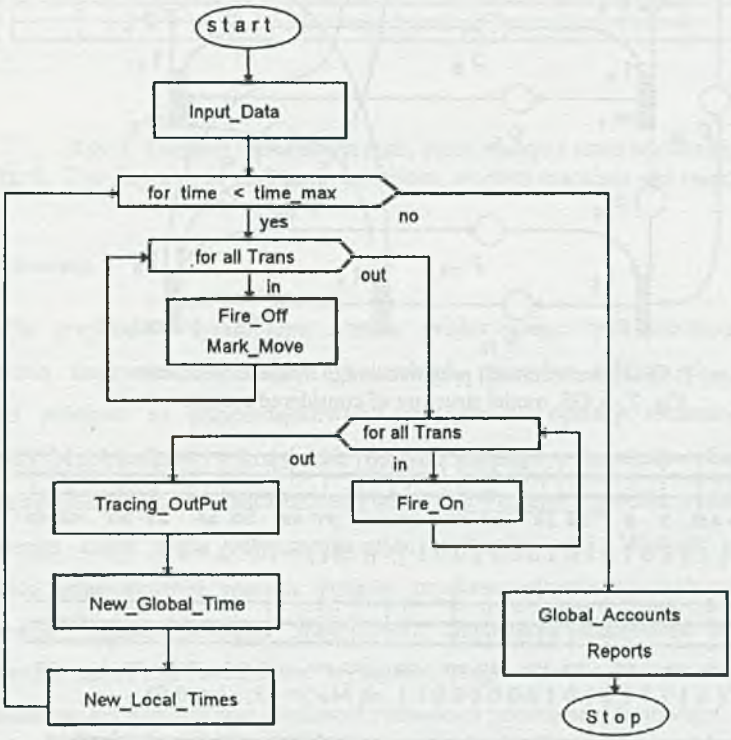
5. Struktura algorytmu symulacji

Algorytm symulacyjny oparty jest na głównej pętli czasu zdyskretyzowanego do obliczanych chwil zmiany znakowań. Znakowanie M_n jest osiągalne ze znakowania M_0 wtedy, jeżeli istnieje sekwencja wzbudzeń $\partial = t_1 t_2 \ldots t_n$, która transformuje M_0 do M_n ($M_0 \xrightarrow{\partial} M_n$). Zbiór wszystkich znakowań osiągalnych z M_0 oznaczamy przez $R(M_0)$. Problem odtwarzalności znakowania polega na rozstrzygnięciu, czy dowolne znakowanie jest osiągalne ze znakowania

początkowego. Problem żywotności dla *ZGS* sprowadzał się do problemu odtwarzalności znakowania.

Określenie nowego stanu w wybranej chwili zachodzi w wyniku wykonania kolejnych procedur:

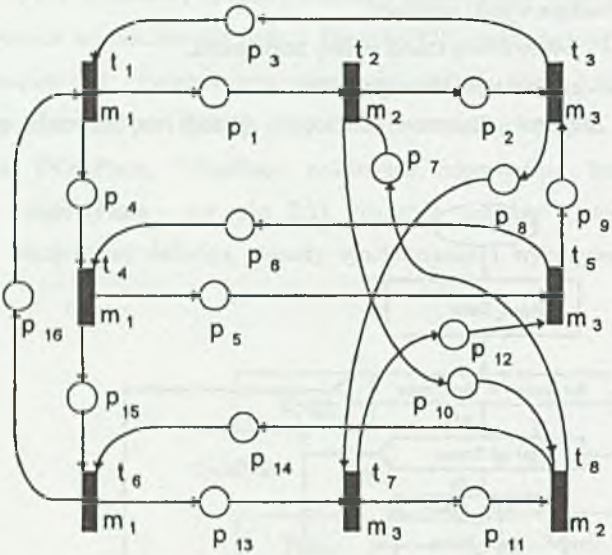
- Fire_Off - zakończenie wzbudzenia przejścia,
- Mark_Move - określenie nowego znakowania osiągalnego,
- Fire_On - wzbudzenie przejścia,
- Tracing_Output - bieżące wyniki symulacji,
- New_Global_Time - wybór nowej chwili zmiany znakowania,
- New_Local_Times - aktualizacja lokalnego czasu,
- Global_Accounts, Reports - końcowe obliczenia i wyniki (współczynniki wykorzystania maszyn).



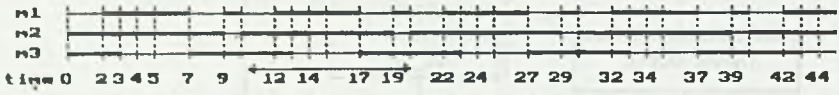
Rys. 6. Diagram sieciowy programu symulacyjnego
Fig. 6. Flow chart of simulation program

6. Przykład obliczeniowy

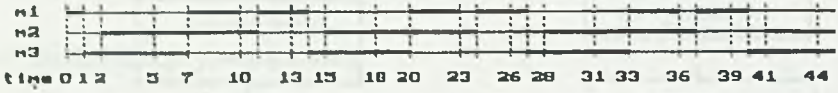
Zastosowanie symulacyjnej metody wyznaczania wartości czasu cyklu odtwarzania stanu znakowania $\gamma(M)$ i wydajności systemu (przepustowości) $\lambda=1/\gamma(M)$ zilustrowano przykładem obliczeniowym. Trójzadaniowe zlecenie produkcyjne wykonywane jest przez trzy różne maszyny. Proces wykonywania zadania może przebiegać w trybie cyklicznym. Struktura przebiegu procesów przedstawiona jest na rys. 7.



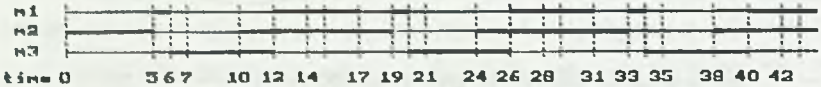
Rys. 7. Graf synchronizacji przykładowego systemu procesów
Fig. 7. A *GS* model structure of considered system



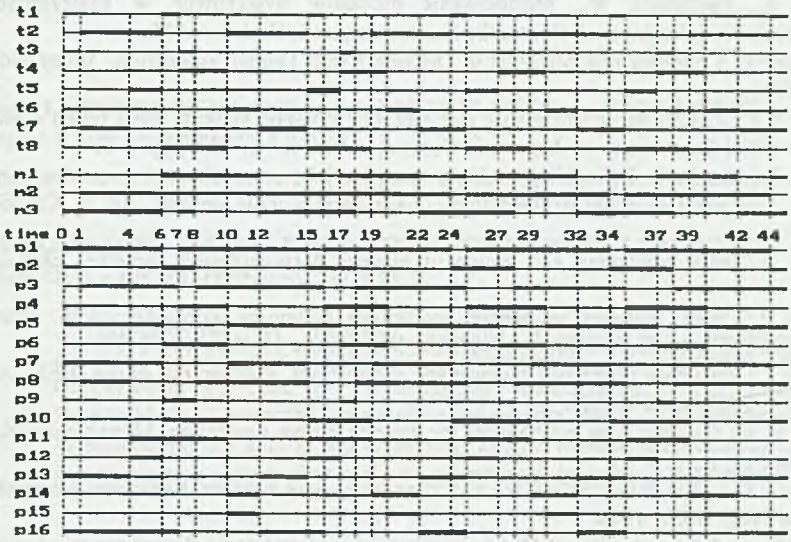
$M_0 = \{1\,1\,0\,0\,1\,0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,1\}$, $\gamma(M_0) = 10$, $\lambda = 0.1$



$M_0 = \{0\,1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,1\}$, $\gamma(M_0) = 13$, $\lambda = 0.08$



$M_0 = \{1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,1\}$, $\gamma(M_0) = 14$, $\lambda = 0.07$



Rys. 8. Diagram wzbudzeń przejść, pracy maszyn i stanu znakowań
Fig. 8. The diagrams of enableness transitions, working machines and marking state

7. Zakończenie

Na przykładzie dyskretnego systemu produkcyjnego typu job-shop rozwiązano zagadnienie sterowania przebiegiem dyskretnych procesów cyklicznych. Wyznaczenie sterowań polegało na przyporządkowaniu zasobów do operacji technologicznych, co zapewniło bezblokadowe wykonywanie operacji każdego z procesów. Zaproponowana technika symulacji wykorzystująca elementy teorii sieci Petriego pozwoliła wyznaczyć wartość minimalnego czasu cyklu odtworzenia stanu znakowań sieci. Wartość ta zapewniała największą przepustowość systemu. Ponadto uzyskano wizualizację zachowania się sieci (wzbudzanie przejść, przepływ znaczników). Stworzenie środowiska programowania obicktowego zapewniło zwiększenie wydajności programowania, panowanie nad dużymi strukturami programowymi oraz możliwość rozbudowy pakietu programowego.

LITERATURA

1. Banaszak Z. (ed.): Modelling and Control of FMS. Petri net approach. Wrocław Technical University Press. Wrocław 1991.

2. Cyklis J., Pierzchała W.: Modelowanie procesów dyskretnych w Elastycznych Systemach Produkcyjnych. Politechnika Krakowska, nr 3, Kraków 1995.
3. Fishman G.: Concepts and Methods in Discrete Event Digital Simulation. Wiley, New York 1993.
4. Hillion H.P., Proth J.M.: Performance evaluation of job-shop systems using timed event-graphs. IEEE Transations on Automatic Control, vol.34, no. 1, 1989, pp. 3-9.
5. Jamroż L., Raszka J.: Scheduling multiple-operation jobs on machines in manufacturing system. Cracow University of Technology. Anniversary issue volume, no. 5, Cracow 1995.
6. Kusiak A. (ed.): Modelling and design of Flexible Manufacturing Systems. Elsevier 1986.
7. Magott J.: Sieci Petriego w ocenie wydajności systemów komputerowych. Prace Naukowe Politechniki Wrocławskiej nr 80, s. Monografie 15, Wrocław 1989.
8. Murata T.: Petri nets: properties, analysis and applications. Proceedings of the IEEE, vol. 77, no. 4, 1989, pp. 541-580.
9. Nance R. E.: On time flow mechanisms for discrete system simulation. Management Sci. 1981, vol 18, no.1, p. 59.
10. Ullman J. D.: Principles of database and knowledge-base systems. Rockville: Computer Science Press, vol.1. 1988.
11. Turbo C++ 3.0 and Object Windows for C++ . User's and Programmer's Guide. Borland International, INC.

Recenzent: Dr hab.inż. Mirosław Zaborowski, prof.Pol.Śl.

Wpłynęło do Redakcji do 30.06.1996 r.

Abstract

This paper is concerned with the problem of both modelling and simulating behaviour of discrete production system (DPS). Some of the main problems appearing in DPS are productivity improvement, performance evaluation and the running comprehension. Such studies can be conducted with the help of modelling and simulating techniques. Our research is focused on design analysis and control at lowest level of DPS. We have considered the type of DPS called job-shop system. The multiple-operation jobs with general precedence constraints are scheduled on multiple machine types. The job is specified as a sequence of machines to visit (i.e. as a routing through the system). Every sequence is defined uniquely for each jobs. The same job can appear several times in the input sequence The objective of scheduling machines is to find the minimum as low as distribution of jobs which leads to the full utilization of the machines.. Considered discrete processes are sequential and cyclic The proceses cooperation facilities is mutual exclusion. We use here the deterministic Petri nets approach to build the simulation model of system. In this model conditions are represented by places and operation by transitions. A transition firing time is associated to time of operation and it is given by a real number. Using known results about marked graph synchronization as a special class of timed Petri nets a performance of DPS is evaluated. The model allows for evaluation minimal cycle time. It enables to estimate such performance measure as a system throughput. Due to known complexity of this type of problem a simulation methodology is presented to obtain solution. The simulation model represents the control procedure ensuring deadlock free cooperation of concurrently flowing processes. It is possible to see the tokens flow through the system how many times a transition fired and analyse how a cetain marking (i.e. reproducible marking, or home state) was reached.