

MODEL PRZEPIYU PRODUKCJI W ZAUTOMATYZOWANYM SYSTEMIE WYTWARZANIA

Przedstawiono strukturę i główne moduły układu sterowania produkcją opartego na skalowalnym modelu macierzowym procesu wytwarzania. Omówiono oprogramowanie umożliwiające samokreowanie skalowalnego modelu macierzowego. Program sterujący produkcją współpracuje z oprogramowaniem Systemu Zarządzania Produkcją, reprezentowanym przez pakiet InTrack™ firmy Wonderware.

MODEL OF PRODUCTION FLOW IN AUTOMATED MANUFACTURING SYSTEM

The structure and main modules of production control system, based on scalable matrix model, are presented. The software for self-creation of scalable matrix model is described. The main production control program co-operates with Manufacturing Execution System, represented by Wonderware® InTrack™.

1. WPROWADZENIE

Znamienną cechą oryginalnego układu komputerowego sterowania produkcją zautomatyzowaną, opracowanego w Katedrze Systemów Wytwarzania Politechniki Krakowskiej, było zastosowanie modelu realizowanego procesu wytwarzania, który był przetwarzany w czasie rzeczywistym. W oparciu o model komputer sterujący przebiegiem produkcji zlecał sterownikom poszczególnych urządzeń kolejne czynności do wykonania, po czym oczekiwał na potwierdzenia ich zakończenia. Układ ten oraz przykład praktycznej jego realizacji przedstawiono m. in. w pracach [1, 2].

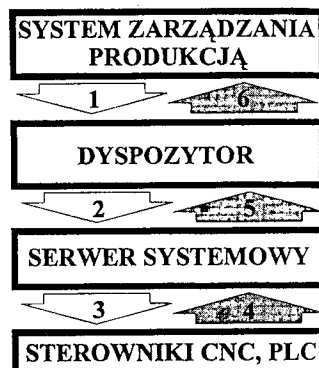
W przypadku zmian asortymentu produkowanych wyrobów należało stosować „nadmiarowy” model po to, aby zmiany te można było uwzględniać bez zatrzymywania sterowanego procesu wytwarzania. Oznaczało to, że do modelu należało wprowadzić wszystkie przedmioty, na które przewidywano zapotrzebowanie w przyjętym horyzoncie czasu. W poszczególnych fazach realizowanego procesu produkcyjnego były wykonywane tylko partie przedmiotów z założonej puli. Pomijając niedogodność i nieracjonalność takiego podejścia, należy zauważyć, że było ono nieskuteczne w dłuższej perspektywie i nie rozwiązywało np. problemu pilnych i niespodziewanych zamówień.

W pracy prezentowany jest układ komputerowego sterowania wytwarzaniem, który umożliwia płynny przepływ realizowanych zleceń produkcyjnych, niezależnie od dynamiki zmian asortymentowych wynikających z bieżącego zapotrzebowania na wyroby. Aby uzyskać tę zdolność opracowano skalowalny model procesu wytwarzania, który może w zależności od potrzeb zmieniać swoje rozmiary (przy usuwaniu przedmiotów zakończonych lub przy wprowadzaniu do produkcji przedmiotów nowych), zachowując przy tym informacje o

bieżącym stanie procesu. Ponadto zapewniono możliwość współpracy komputera sterującego z Systemem Zarządzania Produkcją.

2. STRUKTURA UKŁADU STEROWANIA PRZEPIYWEM PRODUKCJI

Zasadniczą rolę w prezentowanym układzie sterowania przepływem produkcji pełni program *Dyspozytor* (rys. 1). Z *Systemu Zarządzania Produkcją* otrzymuje on dane o zamówieniach, które należy uruchomić (strzałka 1). Przetwarzając *Skalowalny Model Macierzowy (SMM)* procesu wytwarzania wysyła on informacje o tym, którą czynność i kiedy należy rozpocząć. Polecenie rozpoczęcia czynności, w postaci jej nazwy, wysyłane jest poprzez sieć *Ethernet (protokół TCP/IP)* do *Serwera Systemowego* (strzałka 2). W ten sposób *Dyspozytor* zleca do wykonania wszystkie czynności *dopuszczalne* (na danym etapie realizowanego procesu wytwarzania). *Serwer Systemowy* (opisany w pracy [1]) interpretuje otrzymaną nazwę czynności i przygotowuje blok danych „zrozumiałych” dla sterownika urządzenia, które tę czynność będzie wykonywać. Przygotowanie tego bloku wymaga czasem



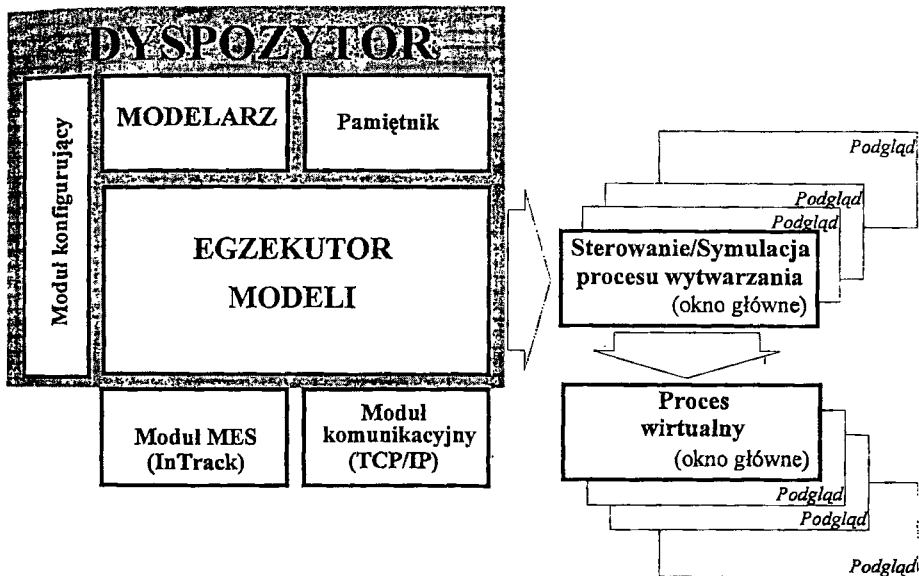
Rys. 1. Struktura układu

informacji pochodzących z modułów systemu wytwarzania wyposażonych w autonomiczne sterowanie (np. z modułów magazynowania, narzędziowego, bazy programów numerycznych). Przygotowany blok danych wysyłany jest przez *Serwer Systemowy* do odpowiedniego sterownika, w sposób dla tego sterownika właściwy, np. poprzez sieć *Ethernet*, lokalną sieć przemysłową, złącze szeregowo (strzałka 3). Po zakończeniu czynności, sterownik urządzenia wykonującego wysyła do *Serwera Systemowego* blok danych potwierdzający jej zakończenie (strzałka 4). *Serwer Systemowy* dopasowuje ten blok do odpowiedniej nazwy czynności i nazwę tę odsyła do *Dyspozytora* (strzałka 5), co stanowi dla niego potwierdzenie pomyślnego zakończenia czynności zleconej do wykonania. Wykonywane jest stosowne przekształcenie *SMM*, a ponadto do *Systemu Zarządzania Produkcją* przekazywana jest odpowiednia informacja umożliwiająca aktualizację bazy danych o stanie zaawansowania sterowanego procesu wytwarzania (strzałka 6).

Jako przykład oprogramowania *Systemu Zarządzania Produkcją* w prezentowanym układzie wykorzystywano pakiet *InTrack 7.0* z zestawu *FactorySuite 2000* f-my *Wonderware*, który jest pierwszym pakietem realizującym zadania zarządzania produkcją pracującym na komputerach klasy PC. Pakiet ten umożliwia m. in. monitorowanie przepływu zleceń produkcyjnych w systemie wytwarzania, wizualizację i nadzorowanie produkcji, archiwizowanie danych i raportowanie, tworzenie historii powstawania wyrobów, przydatnej np. przy poszukiwaniu powodów powstawania braków czy przy wdrażaniu norm serii *ISO9000*. *InTrack 7.0* współpracuje z bazą danych czasu rzeczywistego *InDustrialSQL Server 7.0*, która dzięki integracji z *MS SQL Server* stanowi spójne źródło danych do zarządzania produkcją. *InTrack 7.0* posiada własne moduły komunikacyjne umożliwiające m.in. łączenie z systemami klasy *MRP* (np. *SAP R/3*). Jest wyposażony w szybki system komunikacyjny *SuiteLink* f-my *Wonderware*, a ponadto jest zgodny z protokołami *DDE*, *fastDDE*, *NetDDE*, *OPC*. Obsługuje także mechanizmy *OLE Automation* i *ActiveX*, co pozwala na współpracę z dowolnym oprogramowaniem wykazującym zgodność z tymi mechanizmami.

3. PROGRAM DYSPOZYTOR

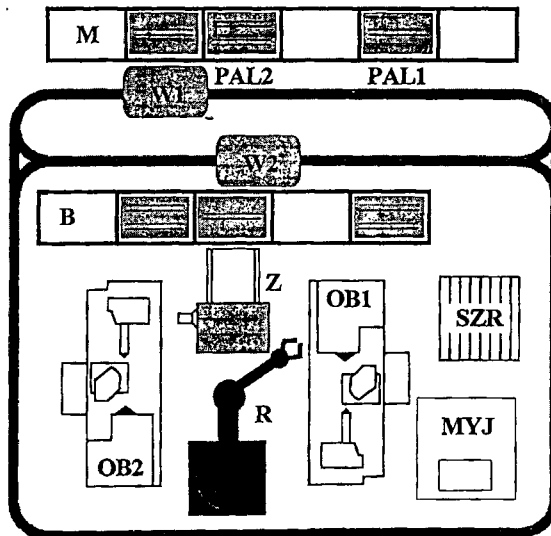
Program Dyspozytor do symulacji i sterowania przepływem produkcji wykorzystuje skalowalny model macierzowy (SMM) systemu wytwarzania i jest 32-bitową aplikacją (rys. 2) pracującą w środowiskach Microsoft® Windows® 95, 98 oraz Windows NT®.



Rys. 2. Struktura programu Dyspozytor

Modelarz służy do wprowadzenia Bazy Reguł, w której za pomocą tzw. szablonów opisane są elementy systemu wytwarzania i ich funkcje. Szablon dla danego elementu składa się z wzorców czynności elementarnych, w których ten element uczestniczy, wraz z przyporządkowanymi im zbiorami wzorców następników, opisującymi porządek czynności elementarnych.

W celu zilustrowania zawartości Bazy Reguł zostanie wykorzystany przykład małego systemu wytwarzania (rys. 3). W systemie tym dwa wózki W1 i W2 dostarczają palety PAL1 i PAL2 z magazynu M do stacji załadowczo – rozładowczej SZR, gdzie są one ładowane przedmiotami. Następnie palety są składane w pięciomiejscowym buforze



Rys. 3. Schemat systemu wytwarzania

B, skąd zabierane są przez zmieniacz Z. Robot R ładuje przedmioty z palety znajdującej się na zmieniaczu na obrabiarki OB1 i OB2, a po obróbce zwraca je na tę samą paletę, która następnie jest oddawana przez zmieniacz do bufora. Z bufora wózki przewożą palety z obrobionymi przedmiotami do stacji mycia MYJ, a następnie do SZR w celu rozładowania, po czym palety są zwracane do magazynu. Poniżej podano szablony dla obrabiarki OB1 oraz dla wózka W1 (część szablonu dotyczącą palety PAL1). Znak \mathcal{E} oznacza następnik wymienionej czynności.

Szablon dla OB1

$_p_;R\gg OB1$ (robot R podaje dowolny przedmiot typu $_p_$ na obrabiarkę OB1)
 $\mathcal{E} _p_;OB1^*$
 $_p_;OB1^*$ (obróbka przedmiotu typu $_p_$)
 $\mathcal{E} _p_;OB1\gg R$
 $_p_;OB1\gg R$ (robot R odbiera przedmiot typu $_p_$ z obrabiarki OB1)
 $\mathcal{E} _p_;R\gg OB1$

Szablon dla W1

$PAL1;M\gg W1$ (pobranie palety PAL1 z magazynu M na wózek W1)
 $\mathcal{E} PAL1;W1\gg SZR$
 $PAL1;W1\gg SZR$ (oddanie palety PAL1 z wózka W1 na stację załadowniczą SZR)
 $\mathcal{E} W1?$
 $_p_;PAL1;SZR\gg W1$ (pobranie palety PAL1 z dowolnymi przedmiotami typu $_p_$ ze stacji SZR na wózek W1)
 $\mathcal{E} _p_;PAL1;W1\gg B$
 $_p_;PAL1;W1\gg B$ (oddanie palety PAL1 z przedmiotami typu $_p_$ do bufora B)
 $\mathcal{E} W1?$
 $_p_;PAL1;B\gg W1$ (pobranie palety PAL1 z przedmiotami typu $_p_$ z bufora B na wózek W1)
 $\mathcal{E} _p_;PAL1;W1\gg MYJ$
 $_p_;PAL1;W1\gg MYJ$ (oddanie palety PAL1 z przedmiotami $_p_$ na stację mycia MYJ)
 $\mathcal{E} W1?$
 $_p_;PAL1;MYJ\gg W1$ (pobranie PAL1 z przedmiotami $_p_$ ze stacji MYJ na wózek W1)
 $\mathcal{E} _p_;PAL1;W1\gg SZR$
 $_p_;PAL1;W1\gg SZR$ (oddanie PAL1 z przedmiotami $_p_$ na stację SZR)
 $\mathcal{E} W1?$
 $PAL1;SZR\gg W1$ (pobranie palety PAL1 ze stacji SZR na wózek W1)
 $\mathcal{E} PAL1;W1\gg M$
 $PAL1;W1\gg M$ (oddanie PAL1 do magazynu M)
 $\mathcal{E} W1?$
 $W1?$ (sprawdzenie gotowości wózka W1)
 $\mathcal{E} PAL1;M\gg W1$
 $\mathcal{E} _p_;PAL1;SZR\gg W1$
 $\mathcal{E} _p_;PAL1;B\gg W1$
 $\mathcal{E} _p_;PAL1;MYJ\gg W1$
 $\mathcal{E} PAL1;SZR\gg W1$
 $\mathcal{E} PAL2;M\gg W1$
 $\mathcal{E} _p_;PAL2;SZR\gg W1$
 $\mathcal{E} _p_;PAL2;B\gg W1$
 $\mathcal{E} _p_;PAL2;MYJ\gg W1$
 $\mathcal{E} PAL2;SZR\gg W1$

Na podstawie szablonu algorytmicznie tworzony jest zbiór wzorcowych cykli elementarnych realizowanych przez ten element, któremu szablon jest przypisany. Poniżej, jako przykład, podano trzy (spośród dziesięciu) wzorcowe cykle elementarne wyznaczone na podstawie powyższego szablonu dla wózka W1.

Cykl 1	Cykl 2	Cykl 3
PAL1;M»W1	<u>p</u> , PAL1; SZR»W1	<u>p</u> , PAL1; B»W1
PAL1; W1»SZR	<u>p</u> , PAL1; W1»B	<u>p</u> , PAL1; W1»MYJ
W1?	W1?	W1?

Szablon opisuje więc wszystkie możliwe cykle pracy elementu, ale w oderwaniu od konkretnych wyrobów, które mogą w tych cyklach brać udział.

Baza Reguł wykorzystywana jest do *samokreowania* SMM. Kluczową rolę w tym procesie odgrywa *dopasowywanie* czynności elementarnych związanych z wykonywaniem konkretnych przedmiotów do ich wzorców. Do wzorca *pasuje* każda czynność elementarna, która w miejscu reprezentowanym przez typ przedmiotu obrabianego (przechowywanego, przemieszczanego) zawiera nazwę przedmiotu tego typu. Na przykład, w powyższym systemie do wzorca p;OB1*, pasuje czynność W323[surowy:po OB1];OB1* związana z obróbką przedmiotu o nazwie W323 (w nawiasach prostokątnych opcjonalnie podaje się stany wyrobów, w tym przypadku zmianę stanu wyrobu W232 z surowego na stan po obróbce na obrabiarce OB1), z kolei czynność P1[gotowy], PAL1; W1 » MYJ (przekazanie gotowych przedmiotów P1 na palecie PAL1 z wózka W1 na stację mycia MYJ) pasuje do wzorca p, PAL1; W1»MYJ. Należy zauważyć, że każda czynność rzeczywista musi pasować do jednego wzorca w szablonie każdego uczestniczącego w niej elementu. Zatem wymieniony w powyższym przykładzie wzorec p, PAL1; W1»MYJ musi wystąpić w szablonach elementów PAL1, W1 oraz MYJ.

Wzorce czynności elementarnych wykorzystywane są także do definiowania *warunków dopuszczalności* czynności ze względu na unikanie zastoju, czyli takich stanów systemu, kiedy nic nie da się zrobić. Przykładem sytuacji prowadzącej do zastoju jest pobranie kolejnych palet z bufora na oba wózki w celu przewiezienia ich na stację mycia, jeżeli stacja jest zajęta. Wtedy wózki nie mogą pozbyć się palet ani też zwolnić stacji. Istota metody unikania zastoju w SMM polega na tym, że pozwala się na rozpoczynanie tylko tych cykli działania urządzeń, które mogą się zakończyć. Warunki dopuszczalności przyporządkowuje się tym wzorcowym cyklom elementarnym, których rozpoczęcie może doprowadzić do zastoju.

Budowę Bazy Reguł omówiono szerzej w pracy [3]. Na jej podstawie dokonywana jest algorytmiczna samokreacja skalowalnego modelu macierzowego, zawierającego tylko te elementy, które biorą udział w realizowanym procesie produkcyjnym.

Głównym modułem programu Dyspozytor jest *Egzekutor Modeli*. Do jego zadań należą:

- tworzenie skalowalnego modelu macierzowego (SMM) realizowanego procesu wytwarzania (na podstawie Bazy Reguł) i przygotowywanie go do przetwarzania w układzie sterowania produkcją;
- skalowanie modelu w przypadku zmian asortymentu produkowanych wyrobów, tzn. dynamiczne - bez przerywania sterowanego procesu - zmienianie jego rozmiarów, w efekcie usuwania bądź dodawania do niego przedmiotów;
- przetwarzanie modelu w trybie symulacji lub sterowania produkcją, polegające na wydawaniu poleceń realizowania odpowiednich czynności przez urządzenia systemu

wytwarzania i odbieraniu od właściwych sterowników CNC i PLC potwierdzeń ich wykonania;

- realizacja tzw. procesów wirtualnych;
- zarządzanie oknami wyświetlającymi informacje o przebiegu symulacji lub sterowania.

Wymieniony wyżej proces wirtualny symuluje działanie systemu wytwarzania począwszy od jego aktualnego stanu i pokazuje mogący zaistnieć, rzeczywisty przebieg zdarzeń. Uruchamiając proces wirtualny w czasie oczekiwania na zakończenie trwających czynności, można np. sprawdzić skuteczność przyjętej heurystyki sterowania.

Przebieg procesu wytwarzania, zarówno rzeczywistego, jak i symulowanego, przedstawiany jest w *oknach modelu*. Okno główne i okna podglądowe mogą się znajdować w kilku trybach wyświetlania: czynności możliwych do wykonania, czynności wykonywanych, stanu zaawansowania produkcji, oraz współczynników wykorzystania maszyn.

4. WSPÓŁPRACA Z SYSTEMEM ZARZĄDZANIA PRODUKCJĄ

Współpraca pomiędzy programem Dyspozytor a oprogramowaniem Systemu Zarządzania Produkcją (ang. Manufacturing Execution System – MES) odbywa się za pośrednictwem *Modułu MES* dołączanego jako biblioteka *DLL*. Zadaniem takiego modułu jest wysyłanie do oprogramowania MES informacji o przebiegu produkcji w systemie. Informacje te przekazywane są modułowi przez program Dyspozytor w formie powiadomień o rozpoczęciu lub zakończeniu wykonywania czynności. Moduł przekształca te informacje do postaci „zrozumiałej” dla oprogramowania Systemu Zarządzania Produkcją.

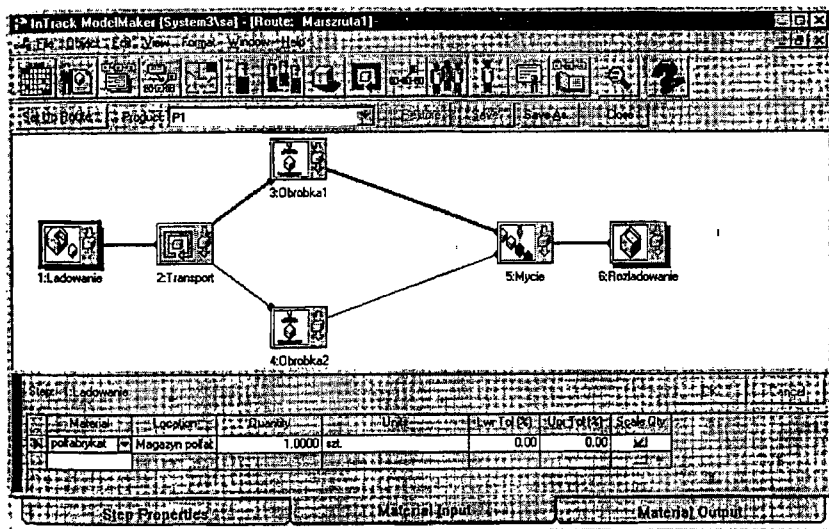
Moduł MES służy ponadto do przekazywania do Dyspozytora informacji o zamówieniach. Wybiera z bazy danych zamówienia zlecone do wykonania i przekazuje do Dyspozytora polecenia rozpoczęcia produkcji właściwych przedmiotów w określonych liczebnościach.

Opracowano bibliotekę *Track.dll*, która jest przykładem Modułu MES, przeznaczonego do współpracy z oprogramowaniem InTrack 7.0 w oparciu o standard *OLE Automation*. Spełnia ona następujące zadania:

1. Udostępnia programowi Dyspozytor (eksportuje) funkcję *InitMES()* dokonującą inicjalizacji modułu oraz funkcję *CloseMES()* dokonującą zamknięcia modułu.
2. Udostępnia funkcję *WorkpieceInfo()* za pomocą której program Dyspozytor przekazuje do modułu informacje o przedmiotach, które znajdują się w aktualnym modelu.
3. Udostępnia funkcję *SelectOrders()*, która pobiera z bazy danych InTrack i zwraca programowi Dyspozytor listę złożonych zamówień, co jest równoznaczne z rozpoczęciem ich realizacji.
4. Udostępnia funkcję *Action()*, którą Dyspozytor wywołuje w celu wykonania w aplikacji InTrack, będącej modelem realizowanego procesu produkcyjnego, właściwej akcji związanej ze zdarzeniem rozpoczęcia lub zakończenia czynności. W przypadku rozpoczynania czynności mogą to być akcje polegające na rozpoczęciu realizacji operacji technologicznej lub pobraniu z magazynu półfabrykatów potrzebnych do rozpoczętej operacji, natomiast w przypadku zakończenia czynności, akcje polegające na zakończeniu operacji technologicznej i przejściu partii w stan oczekiwania przed następną operacją, albo na zakończeniu realizacji całej marszruty technologicznej i umieszczeniu gotowych wyrobów w magazynie.

Trzeba zauważyć, że w prezentowanym układzie sterowania zautomatyzowanym wytwarzaniem występują dwa modele realizowanego procesu produkcyjnego: Skalowalny

Model Macierzowy, przetwarzany przez program Dyspozytor, oraz odpowiednia aplikacja InTrack. Jest oczywiste, że oba modele muszą wykazywać swego rodzaju kompatybilność po to, aby reprezentowały tę samą, choć inaczej przedstawianą, rzeczywistość i aby mogły wymieniać pomiędzy sobą informacje. Do tworzenia aplikacji InTrack polegającego na wykreowaniu modelu procesu produkcyjnego (bazy danych), zawierającego m. in. opis: kalendarza produkcyjnego, klientów, maszyn, materiałów, marszrut i operacji technologicznych, służy program *InTrack Model Maker*. Na rys. 4 pokazano okno tego programu z wykreowaną marszrutą technologiczną wałka P1, który może być obrabiany na maszynie OBI lub OB2.



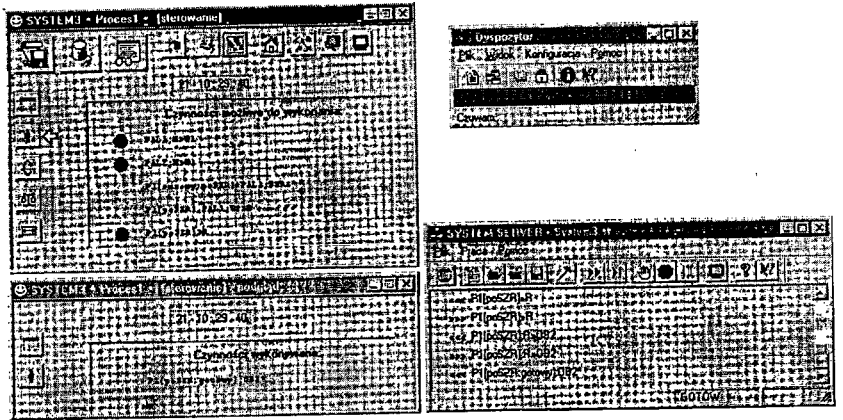
Rys. 4. Okno programu InTrack Model Maker z marszrutą technologiczną wałka P1

5. PREZENTACJA DZIAŁANIA UKŁADU

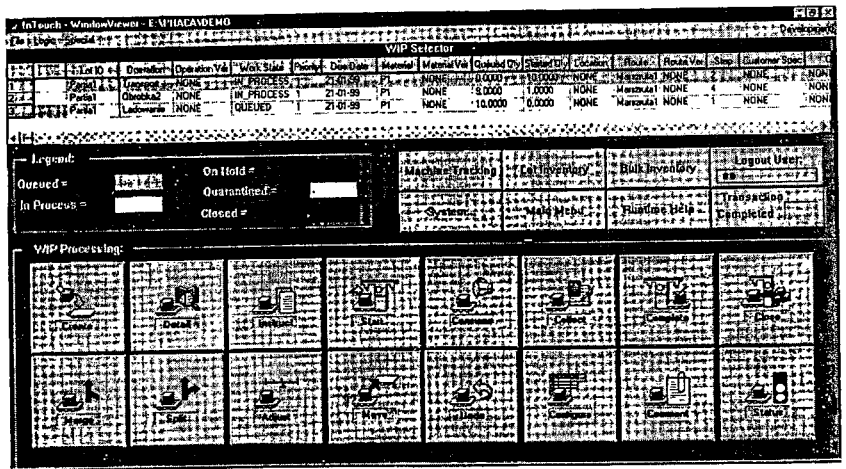
Rys. 5 przedstawia okna programów Dyspozytor oraz Serwer Systemowy w wybranym momencie działania systemu wytwarzania pokazanego na rys. 3 (SYSTEM3). W oknie *Czynności możliwych do wykonania* wyświetlonych jest pięć czynności, z których dwie pierwsze są oznaczone jako niedopuszczalne, ponieważ prowadzą do zastoju. Spośród pozostałych trzech, tzn. : P1[surowy:poSZR]+PAL1;SZR& (ładowanie przedmiotów na paletę), P1[poSZR], PAL1;W2»B (oddanie palety do bufora), i P1[poSZR]»R (pobranie przedmiotu przez robota), z których każda jest "bezpieczna" dla systemu, program preferuje rozpoczęcie tej ostatniej, zgodnie z ustawioną heurystyką sterowania. W oknie *Czynności wykonywanych* wyświetlona jest jedna czynność P1[poSZR:gotowy];OB2* (obróbka przedmiotu P1 na obrabiarce OB2), która trwa. W oknie Serwera Systemowego wyświetlona jest przewijalna lista czynności zleconych do wykonania (oznaczonych «««»), wśród których, jako ostatnia, jest wymieniona trwająca czynność P1[poSZR:gotowy];OB2*, oraz czynności, których pomyślne zakończenie zostało potwierdzone (oznaczone «»»»).

Ta sama sytuacja pokazana jest w oknie *WindowViewer* pakietu wizualizacyjnego *InTouch*, wchodzącego także w skład zestawu *FactorySuite 2000*, który umożliwia utworzenie interfejsu operatora systemu InTrack. *Partial* przedmiotów P1, które są aktualnie obrabiane (Obróbka2) ma symbol *Work State = IN_PROCESS*.

Podczas eksperymentów proces produkcyjny był - z konieczności - symulowany, natomiast w pełni realizowano przepływ informacji symbolizowanych na rys. 1 strzałkami 1, 2, 5 i 6.



Rys. 5. Okna programów Dyspozytor i Serwer Systemowy



Rys. 6. Okno programu InTouch WindowViewer

LITERATURA

- [1] Pierzchała W.: Otwarty układ sterowania zautomatyzowanym wytwarzaniem. Materiały Konferencji AUTOMATION'98, PIAP, Warszawa, marzec 1998, s. 94-101.
- [2] Pierzchała W.: Modernizacja układu sterowania elastyczną linią obróbkową. Materiały Konferencji AUTOMATION'98, PIAP, Warszawa, marzec 1998, s. 287-294.
- [3] Pierzchała W.: Skalujący się model do sterowania przepływem produkcji w zautomatyzowanym systemie wytwarzania. Zeszyty Naukowe Pol. Śl., Seria: Automatyka, z.124, Gliwice, 1998, s. 111-120.