

dr hab. inż. Jerzy Zając, prof. PK  
mgr inż. Adam Kmieciak  
mgr inż. Jarosław Zych  
Politechnika Krakowska

## **ZWIĘKSZANIE REKONFIGUROWALNOŚCI SYSTEMU WYTWARZANIA NA PRZYKŁADZIE CENTRUM PRODUKCYJNEGO TOR**

*Rekonfiguracja systemów wytwarzania jest jednym z najbardziej interesujących tematów badawczych w zakresie systemów sterowania wytwarzaniem. W pracy przedstawiono, na przykładzie Centrum Produkcyjnego TOR, które znajduje się w laboratorium Zakładu Zautomatyzowanych Systemów Produkcyjnych Politechniki Krakowskiej, koncepcję rekonfiguracji czynności elementarnych, aby umożliwić realizację nowych zadań produkcyjnych. Koncepcja wykorzystuje podział czynności elementarnych na akcje złożone z zadań, co umożliwia właściwe rozdzielenie zakresu realizowanych działań pomiędzy część sprzętową i programową wieloagentowego systemu sterowania.*

### **INCREASING THE RECONFIGURABILITY OF MANUFACTURING SYSTEM ON AN EXAMPLE OF TOR PRODUCTION CENTER**

*Reconfigurability of manufacturing systems is one of the most interesting research topics in the area of manufacturing control systems. The paper present on an example of TOR Production Center situated in The Laboratory of Automated Production Systems Department at Cracow University of Technology a concept of online reconfiguration of some elementary activities to satisfy new production requirements. The concept is based on the idea of proper division elementary activities into actions consisting of tasks that allows separating them between logical and physical parts of multiagent manufacturing control system.*

## **1. WPROWADZENIE**

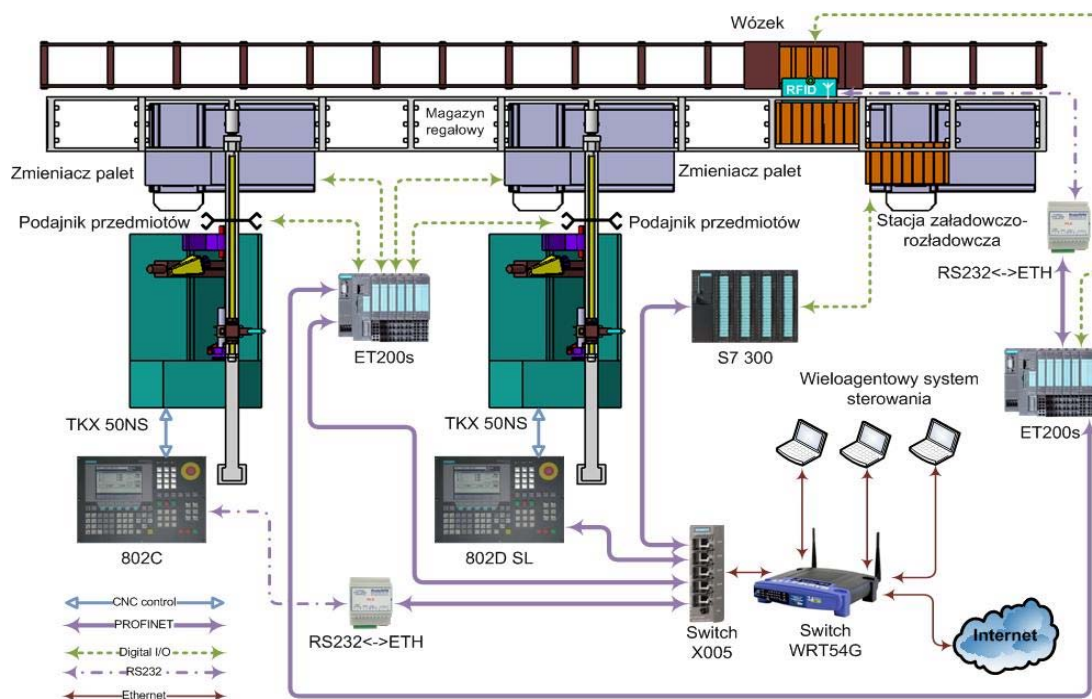
Spośród wielu cech systemów wytwarzania, których znaczenie wynika ze zmieniających się wraz z upływem czasu trendów biznesowych, coraz większą rolę zaczyna obecnie odgrywać rekonfigurowalność. Jest to cecha będąca jakby naturalnym rozwinięciem elastyczności systemów wytwarzania. Elastyczne systemy wytwarzania są „uszytymi na miarę” systemami wyposażonymi w dedykowane podsystemy sterowania, których sposób działania już na etapie jego projektowania został dostosowany do struktury systemu wytwarzania. Rozwój technologii informatycznych oraz automatyki przemysłowej odbywający się w ostatnich latach równolegle ze zmniejszającymi się kosztami ich zastosowania spowodował, że pojawiła się możliwość budowy rozproszonych, rekonfigurowalnych, wieloagentowych systemów sterowania. Zastosowanie takiego podsystem sterowania umożliwia zwiększenie efektywności ekonomicznej przedsiębiorstw poprzez umożliwienie skutecznego ich dostosowywania się do zmian zachodzących na rynku, a wymagających rekonfiguracji posiadanego środowiska produkcyjnego. Rekonfiguracja taka realizowana być może poprzez zmiany w strukturze systemu wytwarzania wynikające np. z dodania dodatkowych maszyn lub urządzeń, co w konsekwencji prowadzić może do zmian w profilu produkcyjnym lub też zapewni skalowalność systemu wytwarzania biorąc pod uwagę jego potencjał produkcyjny.

W pracy omówiono działania dotyczące rekonfiguracji istniejącego systemu produkcyjnego. Nie są związane one jednak ze zmianami w strukturze samego systemu, lecz koncentrują się na innym podziale zadań pomiędzy elementami tworzącymi jego podsystem sterowania, co umożliwia zwiększenie funkcjonalności samego systemu produkcyjnego.

## 2. BUDOWA CENTRUM PRODUKCYJNEGO TOR

Przedstawione w dalszej części artykułu informacje odnoszą się do zmodernizowanego Centrum Produkcyjnego TOR (CP TOR) znajdującego się w Laboratorium Zakładu Zautomatyzowanych Systemów Produkcyjnych Instytutu Technologii Maszyn i Automatyki Produkcji Politechniki Krakowskiej [5].

W skład Centrum Produkcyjnego TOR (rys. 1) wchodzi: *magazyn regalowy MR*, w którym znajdują się palety zawierające produkowane przedmioty; *wózek szynowy WS* przemieszczający palety pomiędzy poszczególnymi stacjami centrum; *stacja załadowczo-rozładowcza SZR* służąca do załadunku półfabrykatów na palety i rozładunku gotowych przedmiotów z palet oraz dwie *stacje obróbkowe SO1 i SO2*.

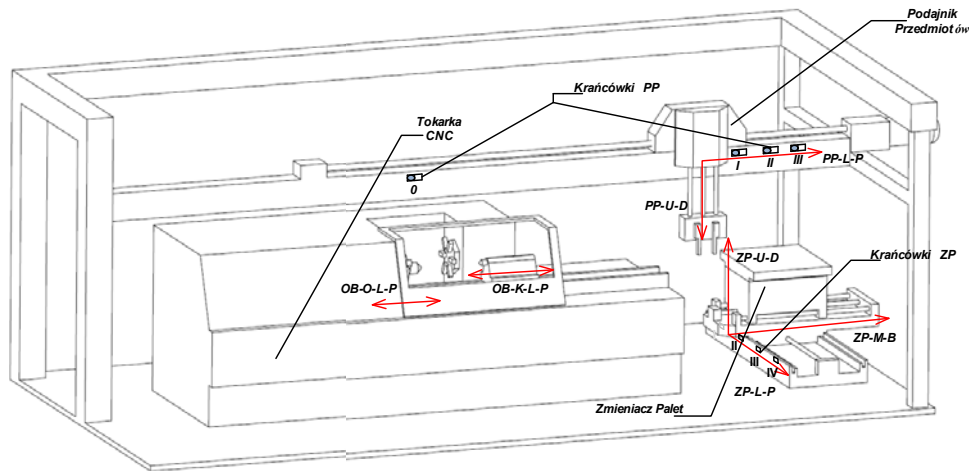


Rys. 1. Schemat strukturalny CP TOR wraz z elementami systemu sterowania

Każda ze stacji obróbkowych zawiera: obrabiarkę (tokarka TKX50NS), zmieniacz palet ZP oraz podajnik przedmiotów PP wyposażony w dwa chwytaki umożliwiające ładowanie i rozładowanie obrabiarki. Dla każdej stacji obróbkowej i stacji załadowczo-rozładowczej występuje dwumiejscowy bufor wejściowo-wyjściowy IN-OUT, z którym z jednej strony współpracuje wózek szynowy, a z drugiej zmieniacz palet.

Przedmiotem szczególnego zainteresowania w niniejszej pracy jest podział zadań w systemie sterowania stacją obróbkową. Na rys. 2 pokazano rozmieszczenie poszczególnych elementów w stacjach obróbkowych. Strzałkami zaznaczono kierunki ruchów, jakie mogą one wykonywać. W zadaniach transportowo-manipulacyjnych uczestniczą kilka elementów

wchodzących w skład stacji obróbkowej. Wykorzystują one do ich realizacji zarówno silniki asynchroniczne napędzające śruby pociągowe (np. podajnik przedmiotów wzdłuż osi PP-L-P, zmieniacz palet wzdłuż osi ZM-L-P) jak i napęd hydrauliczny (np. podajnik wałków PP-U-D).



Rys. 2. Rozmieszczenie elementów stacji obróbkowej

Do wyznaczenia położenia podajnika przedmiotów, zmieniacza palet wykorzystywane są punkty orientacyjne – krańcówki – rozmieszczone wzdłuż osi ich ruchu. Każdy z takich punktów jest jednoznacznie identyfikowany. Poniżej przedstawiono poszczególne zadania transportowo-manipulacyjne oraz elementy stacji obróbkowej, które w nich uczestniczą.

Tabela 1. Zestawienie zadań transportowo – manipulacyjnych dla stacji obróbkowej

Zadanie transportowo-manipulacyjne	Element stacji obróbkowej	Kierunek ruchu wzdłuż osi
Pobranie, oddanie palety	Zmieniacz palet	ZP-L-P
		ZP-U-D
		ZP-M-B
Transport przedmiotów pomiędzy tokarką a zmieniaczem palet	Podajnik Przedmiotów	PP-L-P
Manipulacja przedmiotem w obszarze zmieniacza palet	Zmieniacz palet	ZP-L-P
	Podajnik Przedmiotów	PP-L-P
Manipulacje związane z przedmiotem w obszarze tokarki	Podajnik Przedmiotów	PP-U-D
		PP-L-P
	Tokarka, Konik	OB-K-L-P
Zamknięcie osłony tokarki	Tokarka	OB-O-L-P

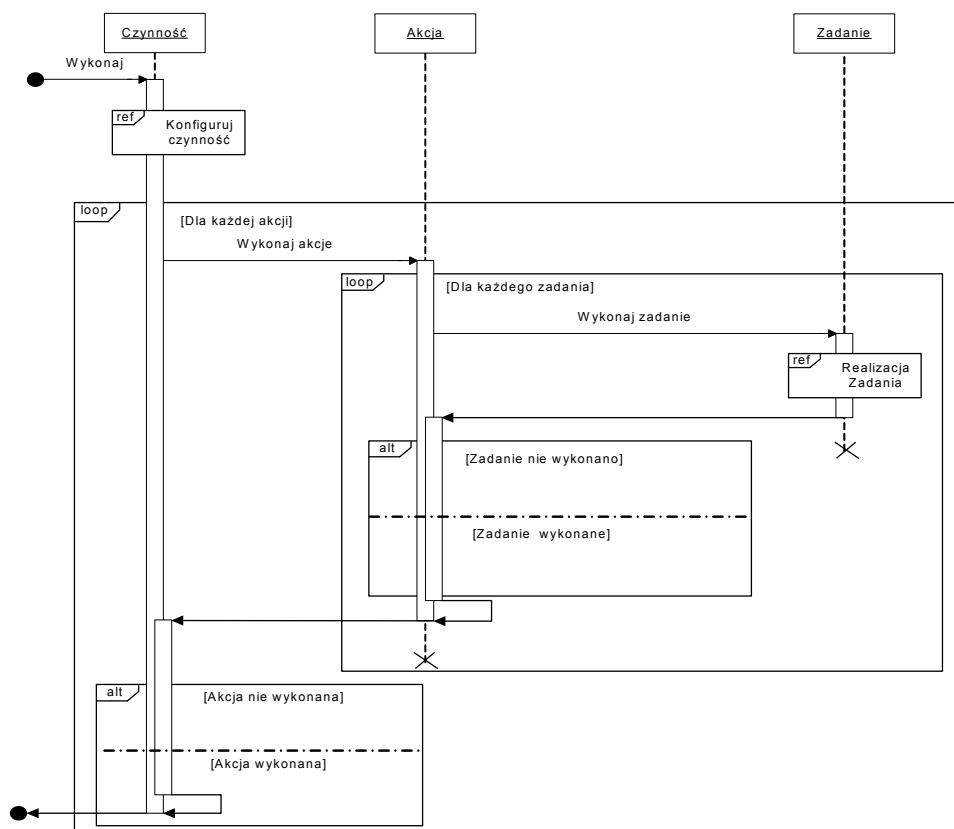
### 3. REKONFIGURACJA CZYNNOŚCI ELEMENTARNYCH W SYSTEMIE CP TOR

Do sterowania centrum produkcyjnym TOR zastosowano wieloagentowy system sterowania AIM [5]. W systemie tym wprowadzono agenta dostosowującego [6] pełniącego rolę warstwy pośredniej pomiędzy sprzętowymi układami sterowania takimi jak PLC czy CNC a częścią logiczną systemu sterowania. Dzięki wprowadzeniu agenta dostosowującego i opracowaniu

zasad podziału czynności na akcje oraz podziału akcji na zadania okazało się możliwe przeniesienie części działań, realizowanych do tej pory poprzez sterowniki PLC, na poziom oprogramowania PC reprezentowanego w tym przypadku przez agenta dostosowującego.

Biorąc pod uwagę procesy planowania i sterowania produkcją, czynność elementarna (operacja) jest działaniem podstawowym. W scentralizowanych systemach sterowania wytwarzaniem wykonanie czynności elementarnej realizowane jest zazwyczaj w całości przez sprzętowe układy sterowania, co oznacza, że wszystkie możliwości jej realizacji muszą być wcześniej przewidziane i oprogramowane. Trudno takie rozwiązanie uznać za otwarte, a wykorzystujący je system za rekonfigurowalny.

Zaproponowane w systemie AIM podejście zakłada, że ze względu na ogólny charakter czynności rozpatrywanych na poziomie logicznym systemu sterowania, czyli takich jak np. pobranie palety, transport palety, obróbka przedmiotów z palety itp., niezbędne jest wprowadzenie podziału czynności elementarnych na zadania elementarne odpowiadające zaprogramowanym wcześniej działaniom sterowników PLC czy CNC, czyli takich jak np. otwarcie szczęk chwytaka czy też uniesienie palety przez zmieniacz palet. W oprogramowaniu AIM założono również, że istnieje możliwość grupowania zadań elementarnych w akcje, takie jak np. pobranie przedmiotu z palety. Oznacza to, że czynności elementarne dzielone są na akcje złożone z zadań. Za poprawne wykonanie czynności elementarnej uznaje się poprawne zakończenie wszystkich akcji wchodzących w jej skład. Z kolei każdą wykonaną akcję uznaje się za zakończoną w przypadku pozytywnego zakończenia każdego z przynależnych jej zadań elementarnych. Algorytm realizacji czynności został przedstawiony na rys. 3 z wykorzystaniem notacji języka UML [2].



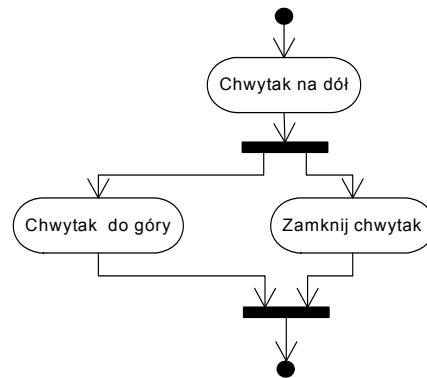
Rys. 3. Diagram sekwencji przedstawiający realizację czynności elementarnej

Konfiguracja czynności i realizacja zadania zostaną omówione szerzej w dalszej części pracy. Scenariusz przedstawiony na rys. 3 zakłada wykonanie kolejnych zadań w ramach akcji. Jeżeli jedno z kolejnych zdań zakończy się wynikiem negatywnym, akcja jest przerywana i uzyskuje ona również wynik negatywny. Jeżeli akcja w ramach czynności *nie wykonała się*, zatrzymywane jest wykonywanie kolejnych akcji. W takim wypadku czynność nie zostaje zrealizowana i operatorowi zgłaszana jest awaria. Z diagramu wynika ponadto, że wynik realizacji zadania elementarnego nie jest przekazywany bezpośrednio do czynności, lecz pośrednio przez akcję. Istnieje ponadto możliwość budowy akcji składających się tylko z jednego zadania elementarnego i czynności składającej się z jednej akcji. Zadania w ramach akcji mogą być wykonywane sekwencyjnie tak ja na powyższym diagramie lub też współbieżnie. Kolejność wykonywania zadań w ramach akcji jest stała i nie ulega zmianom podczas konfiguracji czynności w przeciwieństwie do akcji gdzie kolejność nie jest odgórnie ustalona. Do zapisu akcji wykorzystano w systemie AIM język XML [3]. Listing 1 wraz z umieszczonym obok diagramem aktywności przedstawiają fragment zapisu akcji w przyjętej notacji i ich interpretację graficzną.

```

<action id="3" name="S01-LOOP3">
<description>Oddanie przedmiotu</description>
<ListTask>
  /* wcześniejsze zadania */
  <task id="3" name="CHOPD" description="Otwórz chwytak">
    <nextTask>
      <task id="7" name="CHUPD" description="Chwytak do góry"/>
      <task id="14" name="CHZPD" description="Zamknij chwytak"/>
    </nextTask>
    <previousTask>
      <taskid id="1" name="CHDPD" description="Chwytak na dół"/>
    </previousTask>
  </task>
  <task id="7" name="CHUPD" >
    <nextTask>
    </nextTask>
    <previousTask>
      <task id="3" name="CHOPD"/>
    </previousTask>
  </task>
  <task id="14" name="CHZPD">
    <nextTask>
    </nextTask>
    <previousTask>
      <task id="3" name="CHOPD" />
    </previousTask>
  </task>
</ListTask>
</action>

```



Listing 1. Notacja XML dla zapisu akcji *oddanie przedmiotu*

Listing 1 przedstawia opis akcji *oddanie przedmiotu* realizowanej w ramach czynności obróbka przedmiotów przez stację obróbkową *SO1*. Jest to fragment pełnego XML-owego pliku konfiguracyjnego, który wykorzystywany jest w czasie inicjalizacji agenta dostosowującego *SO1* i dostępny dla procesów konfiguracji i realizacji czynności elementarnej w lokalnej bazie danych. Zawarte w powyższym przykładzie informacje dotyczą listy zadań, w tym poprzedzających (*previousTask*) i kolejnych (*nextTask*) względem zadania definiowanego. Taka forma zapisu sprawia, że w łatwy sposób zapisać można zbiór ruchów poszczególnych elementów stacji bez żmudnej operacji programowania PLC a także nadać im

pożądaną kolejność realizacji. W powyższym przykładzie zadania *chwytak do góry* oraz *zamknij chwytak* wykonywane są współbieżnie.

Zasadniczą cechą opracowanej koncepcji jest to, że procedury niskopoziomowe realizowane są przez sterowniki PLC, natomiast obsługa czynności – zwłaszcza tych konfigurowalnych – realizowana jest na PC. W rozwiązaniu klasycznym obsługa pojedynczej czynności czy nawet grupy czynności realizowana była w całości w sterowniku PLC. W systemie CP TOR do kontroli i wymiany danych pomiędzy sterownikami a agentami dostosowującymi opracowane zostało rozszerzenie darmowej biblioteki libnodave [1]. Do komunikacji wykorzystano rodzinę protokołów TCP/IP. Poniżej przedstawiono fragment kodu w języku Java odpowiedzialnego za połączenie ze sterownikiem PLC oraz odczyt 64 bajtów danych.

```

1.
2. TestISOTCP243 (String host) {
3.     Nodave.Debug=Nodave.DEBUG_ALL;
4.     buf = new char[Nodave.OrderCodeSize];
5.     buf1 = new byte[Nodave.PartnerListSize];
6.     try {
7.         sock = new Socket(host, 102);
8.     } catch (IOException e) {
9.         System.out.println(e);
10.    }
11.    }
12. void run() {
13.     OutputStream oStream = null;
14.     InputStream iStream = null;
15.     byte[] by;
16.     if (sock != null) {
17.         try {
18.             oStream = sock.getOutputStream();
19.         } catch (IOException e) {
20.         }
21.
22.         try {
23.             iStream = sock.getInputStream();
24.         } catch (IOException e) {
25.         }
26.         di = new PLCInterface( oStream, iStream, "IF1",0,
27.             Nodave.PROTOCOL_ISOTCP);
28.         for (int i = 0; i < 3; i++) {
29.             if (0 == di.initAdapter()) {
30.                 System.out.println("Success " + a);
31.                 if (a > 0) {
32.                     for (j = 0; j < a; j++) {
33.                         if (buf1[j] == Nodave.MPIReachable)
34.                             System.out.println("PLC at " + j);
35.                         }
36.                         }
37.                     break;
38.                 }
39.             }
40.         dc = new TCP243Connection(di, 0, slot);
41.         int res = dc.connectPLC();
42.         if (0 == res) {
43.             System.out.println( "Trying to read 64 bytes
44.             (32 words) from data block 1.");
45.             dc.readBytes(Nodave.DB, 1, 0, 64, null);
46.             a = dc.getWORD();
47.         }
48.     }
49. }

```

Listing 2. Kod implementujący odczyt danych ze sterownika PLC (Java)

Obszar wymienianych danych podzielony został na kilka odrębnych części, odpowiedzialnych za przechowywanie danych specyficznych dla różnego rodzaju zastosowań. Poniżej przedstawiono fragment zestawienia danych potrzebnych przy realizacji zadań przez stację obróbkową *SOI*.

Nazwa Zadania:	Obszar Wejściowy			Obszar Wyjściowy			Opis:
	Obszar:	Adres:	Pozycja:	Obszar:	Adres:	Pozycja:	
<b>2. Obrabiarka (OBR)</b>							
OBR-O-K	FLAGS:0	64	6	FLAGS:0	72	6	Otwarcie konika
OBR-C-K	FLAGS:0	64	7	FLAGS:0	72	7	Zamknięcie Konika
OBR-O-O	FLAGS:0	64	4	FLAGS:0	72	4	Otwarcie osłony
OBR-C-O	FLAGS:0	64	5	FLAGS:0	72	5	Zamknięcie osłony
OBR-OBR	FLAGS:0	64	2	FLAGS:0	72	2	Wykonanie programu NC
OBR-L-NC	FLAGS:0	68	-cały bajt	FLAGS:0	brak	brak	Ustawienie numeru Programu
OBR-R-NC	FLAGS:0	64	0		72	0	Potwierdzenie ustawienia Programu
<b>3. Zmieniacz palet (ZMP)</b>							
ZMP-UMG	DB:11	2	2	DB:21		2	Przyjazd zmieniacza palet do magazynu regałowego palet (UMG)
ZMP-U	DB:11	2	0	DB:21		2	Uniesienie palety (PAL) przez zmieniacz palet
ZMP-D	DB:11	2	1	DB:21		2	Opuszczenie palety (PAL) przez zmieniacz palet
UMG-ZMP	DB:11	2	3	DB:21		2	Przyjazd zmieniacza palet z magazynu
ZMP++	DB:11	3	1	DB:21		3	Przesunięcie zmieniacza palet o jedną pozycję w prawo
ZMP--	DB:11	3	0	DB:21		3	Przesunięcie zmieniacza palet o jedną pozycję w lewo
ZMP-P	DB:11	3	3	DB:21		3	Przesunięcie maksymalne zmieniacza palet w prawo
ZMP-L	DB:11	3	2	DB:21		3	Przesunięcie maksymalne zmieniacza palet w lewo

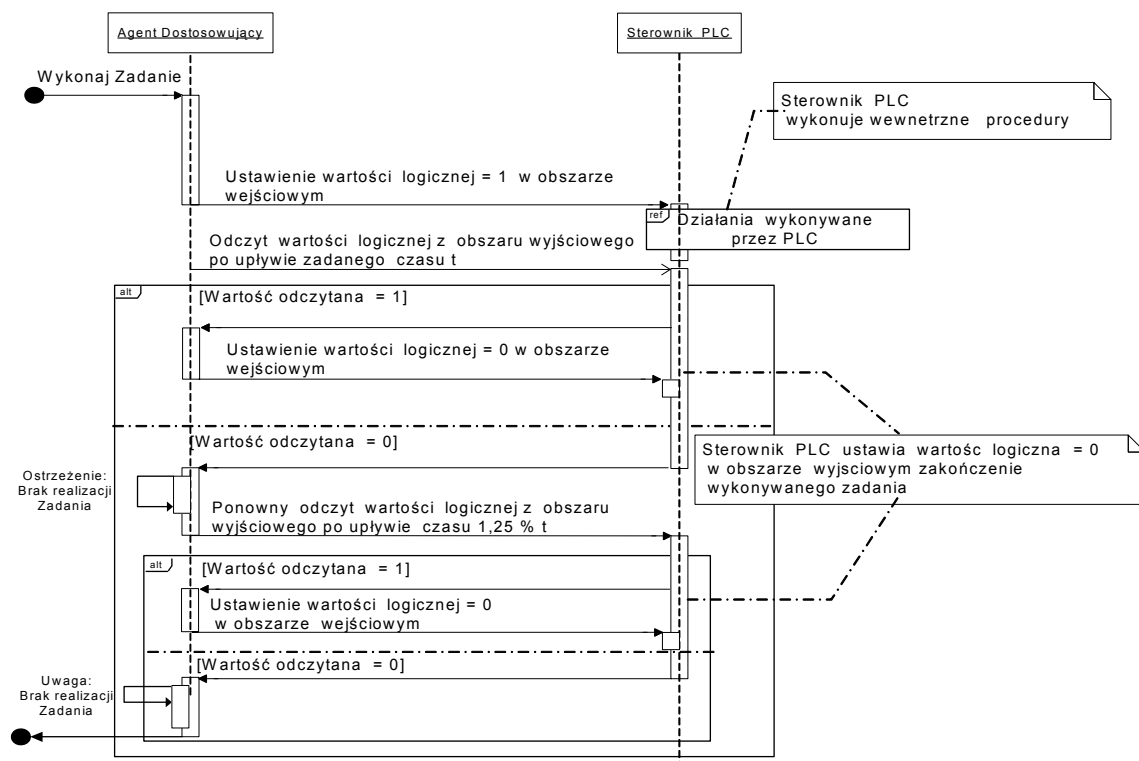
Rys. 4. Fragment zestawienia danych wymienianych pomiędzy agentem dostosowującym a PLC (S7-300)

Przy określaniu obszarów danych podawany jest symbol obszaru (np. DB:11), a położenie danych wewnątrz każdego obszaru danych określone jest przez ich adresy. Adres określa bajt lub słowo wewnątrz obszaru, dodatkowo pozycja określa bit gdzie żądana dana wartość logiczna – 0 lub 1 – jest umieszczona. Dane zostały podzielone na dwa obszary:

- bitów wejściowych używanych do wprowadzania do sterownika sygnałów z otoczenia
- bitów wyjściowych używanych do wyprowadzania rezultatów wykonywanego programu.

Istotnym problemem jest w tym przypadku poprawna i pewna komunikacja podczas wymiany danych. Wymagało to opracowania algorytmu wymiany informacji pomiędzy agentem dostosowującym a sterownikiem PLC. Algorytm ten został przedstawiony na rys. 5 w postaci diagramu sekwencji. Bazuje on na asynchronicznych sygnałach. Wynika to z faktu, iż sterownik PLC jest pasywny i podporządkowany w stosunku do agenta dostosowującego.

W efekcie procesu konfiguracji otrzymujemy zbiór zadań elementarnych, które w przedstawionym przykładzie wykonane są w sposób sekwencyjny. Uruchomienie wewnętrznych procedur w PLC inicjalizowane jest przez ustawienie wartości logicznej równej 1 w zadanym obszarze wejściowym (o ustalonym adresie i pozycji). Dla każdego z zadań oszacowany jest czas  $t$ , w jakim sterownik PLC powinien je wykonać. Po jego upływie oczekuje się wartości logicznej równej 1 w obszarze wyjściowym, który również posiada zdefiniowane parametry (rys. 4). Jeżeli wartość ta jest różna od 1, generowane jest ostrzeżenie oraz ponawiana jest kolejna próba odczytu z obszaru wyjściowego. Jeżeli dalej nie otrzymuje się wartości logicznej równej 1 zadanie uznaje się za nie wykonane i zgłaszana jest awaria. Prawidłowe wykonanie procedur w sterowniku PLC kończone jest ustawieniem wartości logicznej równej 1 w obszarze wyjściowym. Następnie zerowany jest obszar wejściowy przez agenta dostosowującego oraz wyjściowy przez sterownik PLC.



Rys. 5. Algorytm współdziałania pomiędzy agentem dostosowującym a sterownikiem PLC

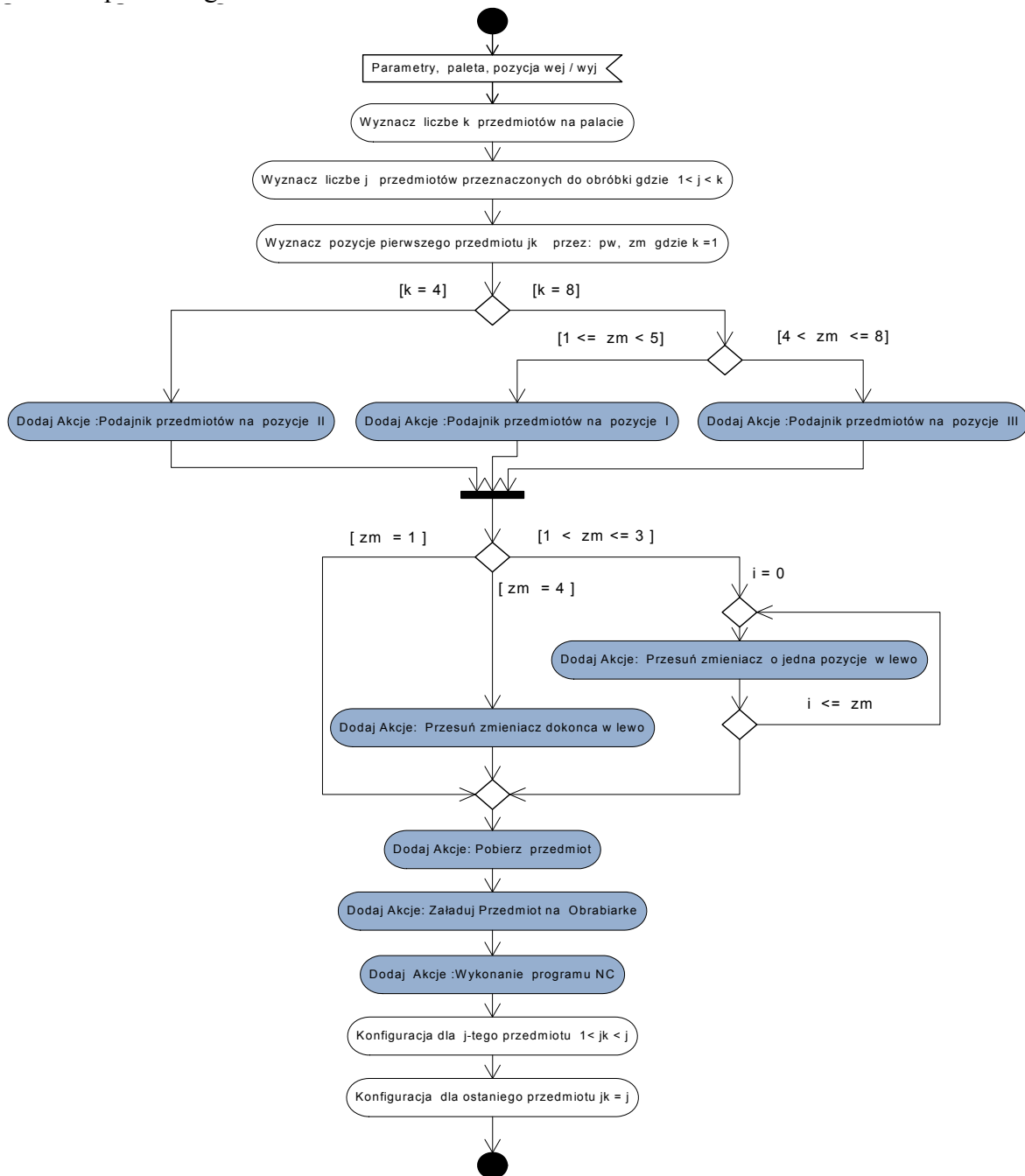
Jako przykład wykonywanych wewnętrznych procedur przez sterownik PLC przedstawiono procedury dla zadania elementarnego jakie wykonuje podajnik przedmiotów przemieszczając się do pozycji wyznaczonej przez krańcówkę dokładną o identyfikatorze II (rys. 2). Najazd na pozycję wyznaczony przez krańcówki o identyfikatorach I, II, III odbywa się zawsze z lewej strony. Każda z krańcówek dokładnych poprzedzona jest krańcówką zwalniającą o takim samym identyfikatorze. Na krańcówkach zwalniających prędkość śruby pociągowej podajnika przedmiotów przełączana jest na wolną i taka pozostaje do momentu najechania na krańcówkę dokładną. Wtedy następuje zatrzymanie ruchu i potwierdzenie wykonania czynności. Napęd podajnika przedmiotów realizowany jest przez silnik asynchroniczny oraz przekładnię dwustopniową. Zmiana kierunku ruchu podajnika przedmiotów wymaga zmiany faz silnika .

Dla przykładowej czynności dane wejściowe: typ palety, liczba przedmiotów przeznaczonych do obróbki na palecie, lokalizacja pobrania i oddalania palety w buforze wejściowo-wyjściowym, w sposób znaczący wpływają na przebieg jej realizacji. Przekłada się to na kolejność wykonywanych zadań elementarnych pogrupowanych w akcje. Na rys. 6 przedstawiono algorytm konfiguracji czynności elementarnej realizowanej przez stację obróbkową *SO1*. Kolorem niebieskim zaznaczono aktywności, podczas których dodawana jest do uporządkowanego zbioru sekwencji akcji kolejna akcja. Informacje dostarczone z części logicznej systemu sterowania pozwalają na określenie liczby *k* przedmiotów na palecie oraz liczby *j* przedmiotów przeznaczonych do obróbki. W systemach sterowania centralnego liczby te były sobie równe. Różnić się mogły jedynie w wyniku pojawienia się braków produkcyjnych lub tzw. „końcówki” zlecenia produkcyjnego.

Przez wyznaczenie liczby *k* możemy określić, z jakim typem palety mamy do czynienia ( $k = 4$  – jednorzędowa,  $k = 8$  – dwurzędowa), co dodatkowo daje informacje o ułożeniu przedmiotów na palecie. Jeżeli liczba  $k = 4$ , podajnik przedmiotów przesuwana się na pozycje



krańcówki o identyfikatorze II (rys. 2.), w przypadku  $k = 8$  o pozycji podajnika decyduje położenie pierwszego



Rys. 6. Sygnały dla rekonfiguracji czynności elementarnej

przedmiotu. Położenie przedmiotów na paletce jest odwzorowane na pozycje podajnika przedmiotów oraz zmieniacza palet zgodnie z ułożeniem ich krańcówek (rys. 2.). Na przykład położenie przedmiotu na paletce dwurzędowej; drugim rzędzie, pierwszy od podajnika przedmiotów, mapuje się na przesunięcie podajnika przedmiotów PP na pozycje II i zmieniacza palet ZP na maksymalne przesunięcie w lewo. Jeżeli pierwszy przedmiot znajduje się w innej pozycji w danym rzędzie (niż wyżej zdefiniowanym) wykonywane są kilkakrotnie akcje *przesuń o jedna pozycje w lewo*. Kolejne dodawane akcje pobierają

przedmiot, transportują nad tokarkę, gdzie jest on obrabiany. Podobnie wygląda konfiguracja – dodawanie akcji dla pozostałych przedmiotów przeznaczonych do obróbki.

#### 4. PODSUMOWANIE

Zastępowanie nieefektywnych ekonomicznie rozwiązań systemów dedykowanych poprzez sprawne, efektywne ekonomicznie, rekonfigurowane systemy modułowe o strukturze rozproszonej jest jednym z trendów rozwojowych współczesnych systemów wytwarzania. Systemy te dzięki swojej modułowości mogą bowiem, w sposób skuteczny reagować na zmiany zachodzące zarówno w otoczeniu systemu produkcyjnego a dotyczące niezbędnych modyfikacji w realizowanym profilu produkcyjnym, jak i zmiany zachodzące w jego wnętrzu a wynikające np. z awarii elementów tworzących system wytwarzania.

Rekonfigurację w systemach wytwarzania można również wprowadzać w istniejących systemach na poziomie pojedynczych urządzeń lub grup urządzeń, tak jak pokazano to w niniejszej pracy. Przedstawione rozwiązanie pokazuje jak dzięki odpowiedniemu podziałowi zadań pomiędzy część sprzętową a programową układu sterowania, powstaje możliwość dostosowywania (w pewnym zakresie) funkcjonalności do aktualnych potrzeb bez konieczności przeprogramowania sprzętowych układów sterowania.

#### 5. LITERATURA

- [1] Libnodave - Exchange data with Siemens PLCs, <http://libnodave.sourceforge.net/>
- [2] OMG UML Resources Page, <http://www.uml.org>
- [3] XML – Extensible Markup Language Resources Page, <http://www.w3.org/TR/REC-xml/>
- [4] Zając J., Chwajół G., Kmieciak A.: Integracja projektowania procesów i sterowania produkcją w zautomatyzowanych systemach wytwarzania. *Pomiary Automatyka Robotyka*, 2/2007, s. 1-9.
- [5] Zając J., Chwajół G., Więk T., Zych J.: Modernizacja systemu sterowania centrum produkcyjnego TOR. *Pomiary Automatyka Robotyka* Nr. 2, 2008, s. 304-310.
- [6] Zając J., Kmieciak A., Słota A., Zych J.: Integracja wieloagentowego systemu sterowania z systemem rzeczywistym na przykładzie CP TOR, *Research Reports Project CII-SK-0030-03-0708 Systems – Equipment - Process SOP'2008*, ISBN: 978-83-7242-481-5. Kraków 2008, s. 416-423.