

BIULETYN TECHNICZNO-INFORMACYJNY

P. 2900/84

TECH

PL ISSN 0239-6645

Nr ind. 35309

10 (268)

1984

Kolegium Redakcyjne:
mgr A. Chróścielewska, dr inż. J. Dyczkowski (redaktor naczelny),
mgr J. Kutrowska (sekretarz redakcji),

Rada Programowa:
inż. J. Bartak, inż. D. Łochocki, mgr S. Majchrzak
mgr inż. A. Musielak, inż. H. Oleksy, mgr inż. H. Piłko,
dr inż. B. Piwowar, dr hab. inż. K. Urbaniec

Warunki prenumeraty

Jednostki gospodarki uspołecznionej, instytucje, organizacje i wszelkiego rodzaju zakłady pracy zamawiają prenumeratę w miejscowych Oddziałach RSW "Prasa-Książka-Ruch", w miejscowościach zaś, w których nie ma Oddziałów RSW – w urzędach pocztowych. Czytelnicy indywidualni opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli. Prenumeratę roczną w cenie 1896 zł należy zamawiać do 25 listopada na rok następny, półroczną do 10 czerwca na II półrocze.



P. 2900/84

**ZRZESZENIE PRODUCENTÓW ŚRODKÓW
INFORMATYKI, AUTOMATYKI
i APARATURY POMIAROWEJ „MERA”**

BIULETYN TECHNICZNO-INFORMACYJNY

Warszawa, październik 1984

SPIS TREŚCI

	Rejestrator zużycia lin stalowych MD 12.....	4
	Miernik WP-1.....	6
A. Kuczora	Rejestrator zużycia lin stalowych MD 12H /defektograf MD 12H/.....	7
J. Kowalczyk	System komputerowy BASF 7/38.....	10
K. Czekala	Technologia mycia pakietów po lutowaniu w wodnych roztworach środków powierzchniowo-czynnych.....	11
J. Makowiecki	"Innowacje" - System konwersyjnego dostępu do banku danych o oprogramowaniu i upowszechnieniu oprogramowania.....	14
A. Aderek	Oprogramowanie zdecentralizowanego systemu automatyki kompleksowej MIR-PROWAY.....	17
	Normatywne planowanie środków produkcji SYNOPS. System wprowadzania i wstępnego przetwarzania danych.....	23
S. Bonkowicz-Sittauer	Propozycja zasad dokumentowania oprogramowania stosowanego w komputerowo wspomaganym projektowaniu.....	27

Opracowanie: Redakcja Biuletynu Techniczno-Informacyjnego "Mera",
 ul. Poezji 19, 04-994 Warszawa /tel. 12-90-11 wew. 17-54/. Wydawca:
 Przedsiębiorstwo Automatyki Przemysłowej "Mera-Pnefal", ul. Poezji 19,
 04-994 Warszawa. Zam.195/84. Nakład 1200 egz.

La technique au service de la paix et du progrès



Le jury du concours pour l'appréciation des meilleurs produits
exposés à la Foire Internationale des Constructions Mécaniques de Brno,
ayant lieu du 12 au 19 septembre 1984, a attribué au produit

Défectographe type MD-12

fabriqué par

MERA STER, Katowice, Pologne

la médaille d'or

aux termes des Règlements pour l'appréciation des produits

J. Jan
Président du jury

Bojda
Secrétaire général du jury

Paul
Directeur général de Polster et Espinasse

REJESTRATOR ZUŻYCIA LIN STALOWYCH MD 12
PRODUKCJI KATOWICKIEGO "MERASTERU"
NAGRODZONY ZŁOTYM MEDALEM
NA MIĘDZYNARODOWYCH TARGACH W BRNIE

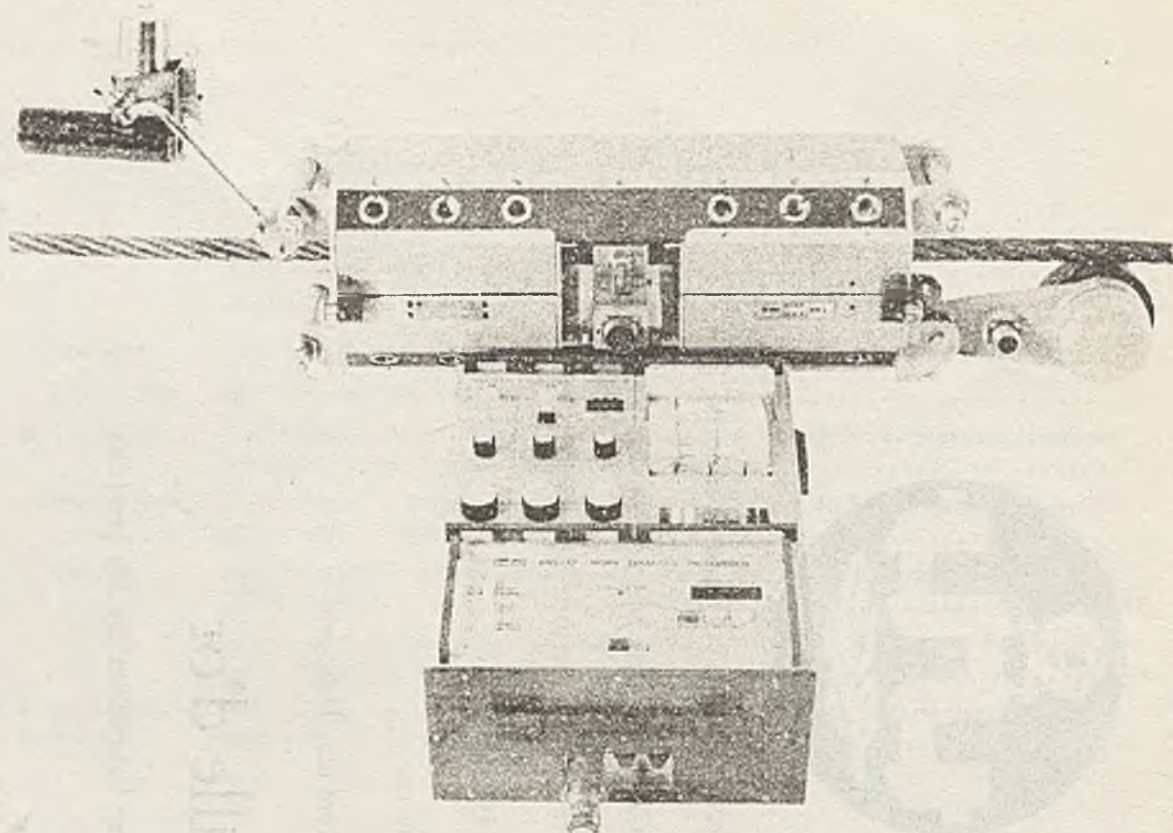


Fig. 1 Rejestrator zużycia lin stalowych MD-12

REJESTRATOR ZUŻYCIA LIN STALOWYCH MD 12

Autorami rejestratora zużycia lin stalowych MD 12 są:

- inż. Józef Kozłowski,
- doc. dr inż. Wojciech Świder,
- mgr inż. Janusz Dziulak,
- mgr inż. Kazimierz Zawada.

Charakterystyka techniczna

1. Przeznaczenie: rejestracja zewnętrznych i wewnętrznych /ukrytych/ wad lin stalowych i stopnia ich zużycia.

2. Czulość: rejestrator pozwala wykryć wady lub zużycie liny począwszy od 0,5-1,0% przekroju odpowiednio do zastosowania głowicy pomiarowej przy nastawianej czulości w zakresie 2 mV/1 mm do 64 mV/1 mm.

3. Liczba kanałów rejestrujących: trzy ± 15 mm każdy z możliwością zmiany funkcji przejścia - rejestracja impulsowa lub rejestracja sumy impulsów w zadanym przedziale całkowania 0,6 do 4 m długości liny.

4. Prędkość pomiarowa przesuwu liny: 0,1 m/s do 3 m/s przy czujniku indukcyjnym oraz 0,0... do 3 m/s przy czujniku hallotronowym.

5. Oznaczenie miejsca wadliwego /zużycia liny/: w zakresie 0-9999 m przez pomiar wg licznika długości liny i przez znacznik wad /SPRAY MARKER/.

6. Zasilanie: 110, 220 V $\pm 10\%$, -15% , 50-60 Hz, 130 VA.

7. Wymiary i waga: 395 x 290 x 150 mm, 12,5 kg.



БЪЛГАРСКА ТЪРГОВСКО
ПРОМИШЛЕНА ПАЛАТА

МЕЖДУНАРОДЕН ПАНАИР-ПЛОВДИВ

ДИПЛОМ

МЕЖДУНАРОДНИЯТ ПАНАИР УДОСТОВЕРЯВА, ЧЕ

Универсален измерителен уред тип ВТ-1

Е НАГРАДЕН С ОТЛИЧИЕТО

ЗЛАТЕН МЕДАЛ

„Ера” - Варшава - ПНР

изложен... на XI Международен есенен технически
панаир - Пловдив, от 24 септември до 1 октомври 1984 г.,
в уверение на което се дава настоящият диплом.

гр. Пловдив, септември 1984 г.

ГЛАВЕН ДИРЕКТОР
НА МЕЖДУНАРОДНИЯ
ПАНАИР - ПЛОВДИВ



ПРЕДСЕДАТЕЛ
НА БЪЛГАРСКАТА ТЪРГОВСКО-
ПРОМИШЛЕНА ПАЛАТА - СОФИЯ

PRZETWORNIKOWY MIERNIK PRĄDU NAPIĘCIA I MOCY WP-1
PRODUKCJI FMIK "ERA" ZDOBYŁ ZŁOTY MEDAL
NA XI MIĘDZYNARODOWYCH TARGACH W PŁOWDIW

MIERNIK WP-1

Autorami miernika WP-1 są:

- doc. dr hab. inż. Czesław Szczepaniak - Instytut Elektrotechniki - Warszawa,
- mgr inż. Krzysztof Stański FMIK ERA,
- mgr inż. Mieczysław Szczepaniak FMIK ERA.

W mierniku znalazły zastosowanie:

1. Patent PRL 100550, "Układ do pomiaru i przetwarzania mocy czynnej", Czesław Szczepaniak.
2. Patent PRL 107045, "Radiator", Krzysztof Stański, Janusz Królikowski.
3. Patent PRL 106708, "Wielozakresowy miernik prądu, napięcia stałego i zmiennego", Mieczysław Szczepaniak.
4. Wzór użytkowy PRL 38193 "Radiator", Krzysztof Stański, Janusz Królikowski.

Dane techniczne

Klasa 0,5 dla prądów stałych i przemiennych
134 zakresy pomiarowe

Znamionowe wartości zakresów pomiarowych:
- prądów

100-200-500 μ A
1-2-5-10-20-50-100-200-500 mA
1-2-5 A

Zakres częstotliwości 30...50...10000 Hz

- napięć

100-500 mV

1-2,5-5-10-25-50-100-250-500 V

Maksymalna częstotliwość od 500 do 10000 Hz
w zależności od zakresu.

Pomiar mocy Od 0,01 mW do 2500 W

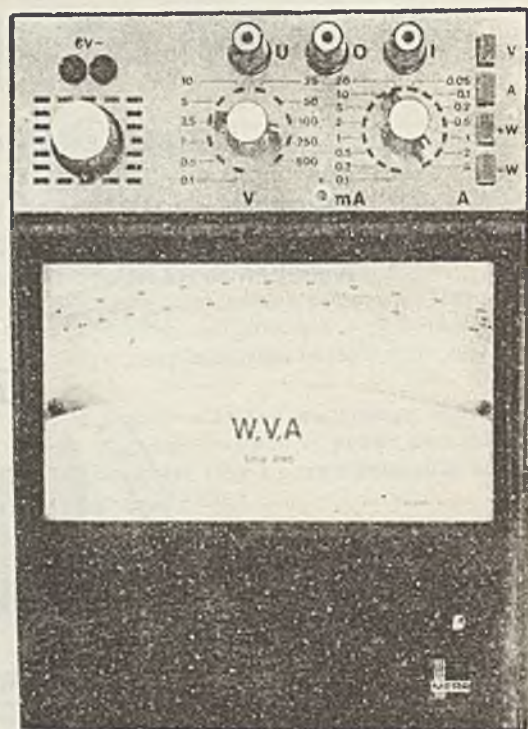
Znamionowe wartości prądów i napięć

100-200-500 μ A
1-2-5-10-20-50-100-200-500 mA-1-2-5 A
0,1-0,5-1-2,5-5-10-25-50-100-250-500 V

Maksymalna częstotliwość od 500 do 5000 Hz
w zależności od zakresu.

Miernik ma 2 gniazda wyjściowe napięcia stałego proporcjonalnego do mierzonej mocy. Przy znamionowych wartościach napięcia, prądu i współczynnika mocy napięcie stałe wynosi 10 V. Stwarza to możliwość rejestracji mocy lub komputeryzacji pomiarów.

Długość podziałki 160 mm
Napięcie probiercze izolacji 2 kV
Masa ok. 4 kg
Wymiary 290x210x120 mm



Fot. 1. Przetwornikowy miernik prądu napięcia i mocy WP-1 zdobył złoty medal na XL Międzynarodowych Targach w Płowdiw/

REJESTRATOR ZUŻYCIA LIN STALOWYCH MD12H /DEFEKTOGRAF MD 12H/

Rejestrator zużycia lin stalowych MD 12H /defektograf MD-12H/ jest zestawem pomiarowym służącym do nieniszczących badań i kontroli zużycia liny. Aparatura ta przeznaczona jest do stosowania podczas rutynowych badań kontroli zużycia lin w miejscu ich eksploatacji, szczególnie lin stosowanych w górnictwie i kolejach linowych. Aparatura opracowywana i produkowana w Centrum Naukowo-Produkcyjnym Systemów Sterowania MERASTER w Katowicach wykorzystuje metodę magnetyczną pola stałego, opracowaną przez Akademię Górniczo-Hutniczą w Krakowie, która badania zużycia lin prowadzi od 1946 r.

Metoda magnetyczna zastępuje wizualne metody badań zużycia lin. Jest to oryginalna polska metoda, mająca na celu lepsze wykorzystanie lin, dzięki uniknięciu przedwczesnej wymiany liny, przy równoczesnym wzroście bezpieczeństwa ich używania. Zastosowanie tej metody umożliwia wykrycie wewnętrznych i zewnętrznych wad i zużycie liny w warunkach jej eksploatacji bez potrzeby jej niszczenia i rozplatania.

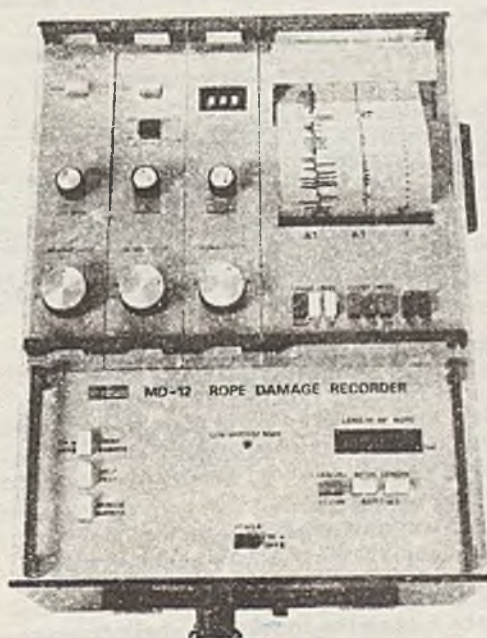
Polska metoda badania magnetycznym polem stałym jest jedną z kilku nowoczesnych metod stosowanych na świecie. Wykazuje nawet nad nimi przewagę, gdyż zapewnia uzyskanie większej ilości informacji o stanie zużycia badanej liny. Wykazały to między innymi badania porównawcze przeprowadzone przez brytyjski Urząd Bezpieczeństwa Górniczego w Sheffield na zamówienie agencji EWG. W Polsce metoda magnetyczna jest już powszechnie stosowana w górnictwie węgla i rud oraz kolejach linowych. Powstała nawet norma branżowa BN 79/5021-09 "Liny kopalniane wyciągowe okrągłe". Oznaczenie stopnia zużycia metodą magnetyczną, która jest pierwszym w świecie tego rodzaju dokumentem, oraz zalecenie Ministerstwa Górnictwa badania lin aparaturą opartą na metodzie magnetycznej, produkowa-

ną między innymi przez Centrum Naukowo-Produkcyjne Systemów Sterowania w Katowicach.

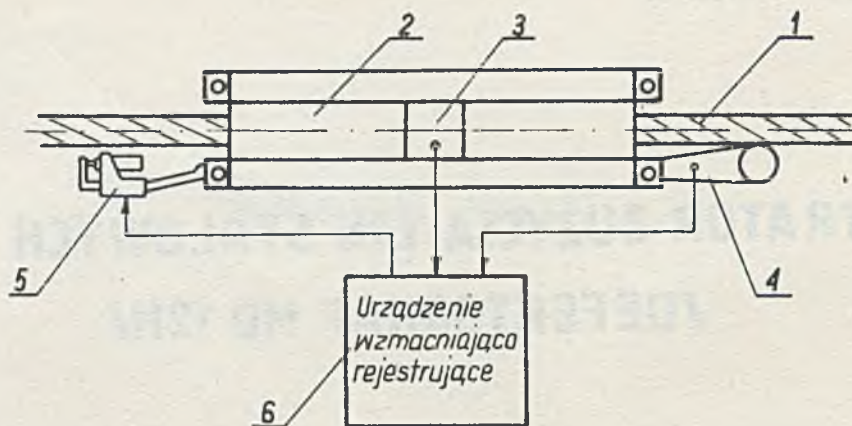
Zestaw pomiarowy rejestratora zużycia lin stalowych MD 12H

Rejestrator zużycia lin stalowych MD 12H jest zestawem pomiarowym, w skład którego wchodzi:

- głowica pomiarowa z przetwornikiem długości liny, którą zakłada się na badaną linę. Głowica jest źródłem sygnałów z zamontowanego w niej czujnika indukcyjno-hallotronowego, a



Fot. 1.



Rys. 1. Podstawowy zestaw pomiarowy rejestratora zużycia lin stalowych MD-12: 1. badana lina, 2. magnes trwały głowicy pomiarowej, 3. czujnik głowicy pomiarowej, 4. przetwornik długości liny, 5. znakownik spryskiwacz, 6. defektograf lub defektograf z adaptorem sygnałów hallotronowych lub defektoskop.

Przetwornik długości liny dostarcza sygnały umożliwiające określenie prędkości ruchu liny, kierunku ruchu oraz pomiar przesunięcia, defektograf rejestrujący na taśmie termoczułej lub taśmie wykresowej sygnały z czujnika pomiarowego oraz zawierający układy umożliwiające sumowanie uszkodzeń na nastawialnych odcinkach długości liny, odpowiedni przewód taśmy, obsługę czujnika wykorzystującego ewki indukcyjne i hallotrony, jako elementy pomiarowe.

• dodatkowym urządzeniem wchodzącym w skład zestawu jest znakownik /SPRAY MARKER/, który służy do znakowania farbą miejsc na linie, w których wykryto uszkodzenia.

Zasada działania rejestratora zużycia lin stalowych MD 12H /defektografu MD 12H/

Podczas przesuwu liny przez głowicę pomiarową następuje jej namagnesowanie wzdłuż odcinka liny wewnątrz głowicy pomiarowej. Uszkodzenia liny powodują zmiany w rozkładzie pola magnetycznego w głowicy, wywołując sygnały elektryczne w czujniku głowicy.

Uszkodzenie lin zmieniające jej przekrój ferromagnetyczny można podzielić na:

- lokalne bądź skokowe zmiany przekroju liny /złamane druty, gniazda korozji/,
- długie zmiany przekroju /równomierna korozja i starcia drutów/,
- skokowe zmiany przekroju przechodzące w długie zmiany /zapłoty, brak odcinka drutu, itp. /.

Sygnały z czujnika indukcyjno-hallotronowego, który zastosowano w tym zestawie można podzielić na dwa sygnały:

- Sygnał z przetwornika indukcyjnego, tj. si-

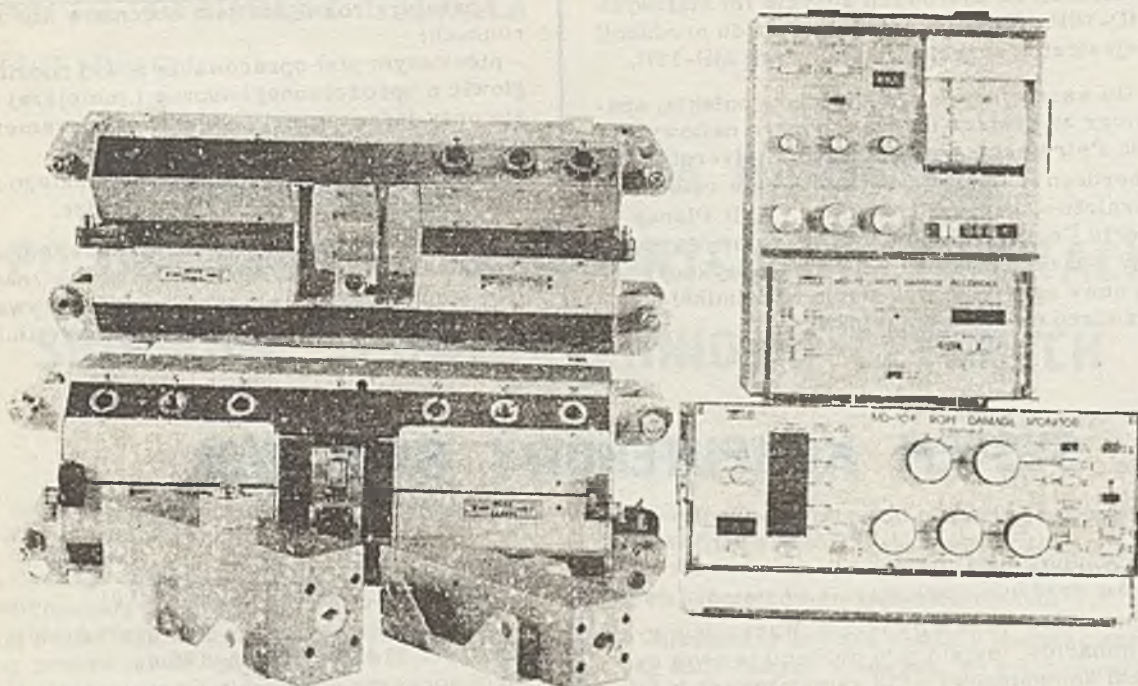
ła elektromotoryczna indukowana w cewce pomiarowej, której wartość i kształt pozwala na określenie wielkości i rodzaju uszkodzenia. Sygnał ten uzyskiwany jest tylko przy względnym ruchu liny i czujnika. Daje bardzo dobre wyniki przy wadach liny typu lokalnej lub skokowej zmiany przekroju liny.

- Sygnał z przetwornika hallotronowego, tj. napięcie Halla, którego wielkość jest proporcjonalna do składowej stycznej indukcji; sygnał ten jest niezależny od prędkości przesuwu liny, umożliwia wykrywanie i określanie długich zmian przekroju liny.

Sygnały otrzymywane z głowicy pomiarowej zapisywane są na taśmie termoczułej, która jest dokumentem i rezultatem badania. Specjalista na podstawie zapisu znajduje miejsce osłabienia liny w funkcji jej długości i położenie w przekroju liny, a następnie wyznacza z dużą dokładnością stopień osłabienia liny w wyniku jej zużycia i może w sposób autorytatywny zakwalifikować linę do jej dalszej eksploatacji lub zażądać jej wycofania.

Rejestrator zużycia lin stalowych MD 12H jest oryginalnym rozwiązaniem opracowanym w MERASTER. Charakterystyczne cechy tego urządzenia, które świadczą o jego nowoczesnym rozwiązaniu i zdecydowały m. in. o przyznaniu mu Złotego Medalu 26 Międzynarodowych Targów Maszynowych w Brnie to:

- zastosowanie czujnika indukcyjno-hallotronowego dającego wszystkie informacje do oceny stanu zużycia badanej liny,
- nowoczesnie rozwiązany specjalizowany rejestrator z zapisem wad i zużycia liny na papierze termoczułym,



Fct. 2.

- synchroniczny przesuw taśmy wykresowej z przesuwem liny,
- kompensacja wpływu prędkości liny w zakresie od 0,1 - 3 m/s.,
- pomiar długości badanej liny do 9999 m.
- zapis sumujący wady umożliwiający oznaczenie osłabienia liny w miejscach zagęszczenia uszkodzeń,
- możliwość przesuwu papieru ze stałą nastawialną prędkością,
- możliwość znakowania farbą wykrytych miejsc uszkodzeń liny.
- możliwości badania lin o średnicach od 8 mm do 85 mm przy pomocy rodziny głowicy typu GP1 - GP6 /produkcji MERASTER/,
- urządzenie zwiększające w sposób pośredni bezpieczeństwo ludzi korzystających z urządzeń wyciągowych, zarówno w kopalniach węgla, rud czy kolejach linowych.

Rejestrator zużycia lin stalowych MD 12 jest urządzeniem w całości wykonanym w technice analogowo-cyfrowej z elementów produkcji polskiej oprócz jednego elementu, jakim jest przetwornik cyfrowo-analogowy K572PA1 produkcji ZSRR.

Inne urządzenia do badania lin produkcji MERASTER

Należy zaznaczyć, że rejestrator zużycia lin stalowych MD 12H nie jest jedynym urządzeniem produkowanym przez Centrum Naukowo-Produkcyjne Systemów Sterowania w Katowicach do badania lin stalowych. Wcześniej opracowanym urządzeniem jest unikalny przyrząd przeznaczony do szybkiej i łatwej kontroli stanu lin anali-

zator cyfrowy zużycia lin stalowych MD 10P /defektoskop MD-10P/. Kompatybilny do głowic rodziny GP tworzy z nimi zestawy do badania lin, które przez prostotę ich obsługi oraz czytelny wynik mogą być używane do częstych szybkich kontroli lin z natychmiastowym odczytem wyniku badania. Zestaw taki, poprzez częste statystyczne badania, może służyć do prognozowania czasu zużycia liny. Jego działanie polega na klasyfikacji uszkodzeń według amplitudy i zliczaniu ich. Klasyfikacji dokonuje się w pięciu przedziałach, które ustawiane są w taki sposób, że 5 przedział zlicza 100% dopuszczalne uszkodzenia liny, a kolejne niższe 80%, 60%, 40%, 20%. W ten sposób rezultat kontroli stanu liny jest bezpośrednio odczytywany z liczników uszkodzeń. Oprócz zliczania pojedynczych wad liny możliwe jest zliczanie zsumowanych /ścalkowanych/ na nastawialnych długościach lub w nastawianych odcinkach czasu zagęszczonych uszkodzeń liny.

Analizator cyfrowy zużycia lin stalowych MD 10P wyposażony jest w wyjście analogowe umożliwiające współpracę jego z typowym rejestratorem lub magnetofonem pomiarowym. Może współpracować z drukarką alfanumeryczną i drukować wyniki pomiaru. Możliwe jest także wykorzystanie go, jako inteligentnej końcówki do współpracy z komputerem.

W Polsce większość kopalń wyposażonych jest w aparaturę do badania lin produkcji MERASTER są to: rejestrator zużycia lin stalowych MD-8 na licencji AGH, który początkowo był produkowany w MERASTERZE, potem w zmodernizowanej wersji tego rejestratora zwanej reje-

stratorem zużycia lin stalowych MD-8M oraz analizatorów cyfrowych zużycia lin stalowych MD-10P. Obecnie wdrażany jest do produkcji rejestrator zużycia lin stalowych MD-12H.

Do ważniejszych użytkowników polskiej aparatury za granicą należą: concern naftowy British Petroleum, Postal Board, University of Aberdeen w Wielkiej Brytanii, całe rumuńskie górnictwo, czeskie górnictwo, Walt Disney World Co. w USA, Ponadto aparaturę tą zakupiły Brazylia i Francja. Obecnie wykonywane są nowe specjalne wykonania tych unikalnych urządzeń na rynki angielskie.

Plany rozwoju aparatury produkcji MERASTER

Aparatura rozwijana jest obecnie w kilku kierunkach:

- pierwszym jest opracowanie nowej rodziny głowic o uproszczonej budowie i mniejszej masie przy jednoczesnej poprawie jej parametrów pomiarowych,
- drugi to opracowanie ręcznego, lekkiego testera uszkodzeń lin o prostej obsłudze.

Następnym krokiem będzie opracowanie mikroprocesorowego urządzenia, które w znacznym stopniu zastąpi eksperta w opracowywaniu wyniku pomiaru, chociaż go nie wyeliminuje.

SYSTEM KOMPUTEROWY BASF 7/38

We wrześniu, październiku i listopadzie br. firma BASF przeprowadza w Warszawie cykl seminariów, na których prezentuje swój system komputerowy 7/38 zainstalowany w Ośrodku Obliczeniowym Uniwersytetu Warszawskiego.

System BASF 7/38 jest w pełni kompatybilny z systemami IBM serii 370 i może wykorzystywać następujące systemy operacyjne: DOS/VS, OS/VS1, DOS/VSE, OS/VS2 z ograniczeniami oraz VM/370.

Podstawowe charakterystyki systemu są następujące:

- rozszerzalność pamięci do 8 MB,
- 16 MB pamięć wirtualna,
- możliwość dołączania do 8 kanałów, cztery z nich mogą być blokowo-multipleksorowymi,
- szybkość wewnętrzna procesora 85 ns.

- cykl pamięci głównej 360 ns,
- pamięć buforowa na elementach ECL o pojemności 32 KB z cyklem 170 ns,
- równoczesne pobieranie i wykonywanie instrukcji,
- 8 B szerokość dostępu z CPU i kanałów do głównej pamięci,
- do 288 podkanałów, 32 z nich dzielonych,
- do 512 podkanałów dla bajtowo-multipleksorowanych kanałów,
- adapter kanał-kanał.

Firma kupującemu daje gwarancję na dwa lata, personel przeszkolony jest w ośrodku w Ludwigshafen. W Polsce firma organizuje skład konsygnacyjny części zamiennych. Na seminarium przedstawiciele firmy BASF podali konkretne ceny wszystkich modułów systemu.

Porównanie systemów BASF 7/38, IBM/370-158-3 i EC 2055 przedstawiono w tabeli

Cecha	BASF 7/38	IBM/370-158-3	EC 2055
Wydajność względna	45	45	25
MPS	1.0	1.0	0.45
Pamięć główna:			
- wymiar /MB/	1 do 8	1 do 6	1 do 2
- przyrosty /MB/	1	1	1
- drogi dostępu	2	2	1
Pamięć buforowa b KB	32	16	
Cykl pracy /ns/	85	115	400
Ilość kanałów	1 do 8	1 do 6	1 do 5
Sterowanie mikroprogramowe	TAK	TAK	TAK
Powierzchnia instalacyjna m ²	3.7	4.4	
Zasilanie wymagane KVA	16	24	
Waga /KG/	2.000	2.600	

mgr inż. JOANNA KOWALCZYK

mgr inż. KAZIMIERZ CZEKAŁA
ZE "ELWRO"

TECHNOLOGIA MYCIA PAKIETÓW PO LUTOWANIU W WODNYCH ROZTWORACH ŚRODKÓW POWIERZCHNIOWO-CZYNNYCH

Na wstępie niniejszego artykułu należy postawić dwa pytania:

- dlaczego należy myć pakiety i podzespoły elektroniczne po lutowaniu?
- jakie do tego celu trzeba stosować środki, aby proces mycia był skuteczny?

Odpowiedź na powyższe pytania, zwłaszcza na drugie, nie może być prosta ani jednoznaczna. Proces mycia pakietów i podzespołów po lutowaniu, uwarunkowany jest przede wszystkim konstrukcją wyrobów, wielkością produkcji oraz stosowanymi w procesie lutowania materiałami.

Dlaczego należy myć pakiety i jakie stosować środki

Pozostałości topników żywicznych i kalafonicznych nie usunięte z płytek drukowanych i podzespołów powodują, ze względu na obecność związków o charakterze jonowym /aktywatory topników lub produkty ich rozkładu/, korozję połączeń lutowanych i zmniejszenie rezystancji. Podstawowe składniki topników, tj. kalafonia lub inne substancje żywiczne, pomimo że same nie powodują korozji powinny być całkowicie usunięte, gdyż mogą być przyczyną:

- obniżenia rezystancji w wyniku gromadzenia kurzu lub absorpcji wilgoci,
- złej adhezji impregnatów lub powłok ochronnych,
- wadliwej pracy przełączników.

Do skutecznego usuwania pozostałości topników po lutowaniu można stosować różnego typu mieszaniny rozpuszczalników organicznych. Mieszanina taka, aby była efektywna, powinna zawierać niepolarny rozpuszczalnik organiczny, usuwający niepolarne składniki topników i olej lutowniczy oraz polarny rozpuszczalnik zdolny do usuwania jonowych pozostałości, takich jak: aktywatory topników i produkty ich rozkładu. Niepolarnymi składnikami mieszanin myjących są najczęściej chlorowane lub fluorowane pochodne etanu. Jako polarne dodatki stosuje się niskocząsteczkowe alkohole, takie jak: alkohol metylowy, etylowy, propylowy lub izopropylowy. Polarne i niepolarne skła-

dniki środków myjących powinny efektywnie usuwać pozostałości topników lecz nie mogą powodować uszkodzeń elementów elektronicznych lub opisów na tych elementach.

Mieszaniny myjące powinny spełniać szereg dodatkowych wymagań, do których należy zaliczyć,

- stabilność składu w warunkach pracy,
- nietoksyczność,
- niepalność,
- niskie straty wskutek odparowania,
- możliwość regeneracji,
- niska cena.

Do idealnego środka myjącego zbliżone są mieszaniny oparte na 1, 1, 2 - trójfluoro, 2, 2, 1 trójchloroetanie /freon 113/ lub 1, 1

dwufluoro, 1, 2, 2, 2, czterochloroetanie /freon 114/. Jedynym mankamentem tych mieszanin myjących jest ich wysoka cena. Mieszaniny myjące zawierające freon 113 lub 114 produkowane są w wielu krajach Europy Zachodniej, USA i Japonii. Nie są one produkowane w Polsce, ani też w krajach Rady Wzajemnej Pomocy Gospodarczej.

Alternatywnym rozwiązaniem są mieszaniny myjące będące wodnymi roztworami środków powierzchniowo-czynnych. Składy tych mieszanin produkowanych w krajach Europy Zachodniej, USA i Japonii pod różnymi nazwami handlowymi nie są ujawnione przez producentów. Użycie wodnych roztworów środków powierzchniowo-czynnych do mycia pakietów po lutowaniu uwarunkowane jest hermetycznością stosowanych elementów i podzespołów elektronicznych oraz całkowitą rozpuszczalnością w wodzie stosowanych materiałów w procesie lutowania, tj. topników oraz olejów lutowniczych. Mieszaniny myjące oparte na wodnych roztworach środków powierzchniowo-czynnych spełniają wymogi, jakie stawia się środkom do mycia pakietów i podzespołów elektronicznych. Istotną ich zaletą w stosunku do mieszanin opartych na freonach jest znacznie niższa cena. Ze względu na brak krajo-

wych, jak i niedostępność w krajach RWPG środków do mycia pakietów i podzespołów elektronicznych, wzrastające trudności z ich importem z II obszaru płatniczego, na początku 1980 r. podjęto w ZE ELWRO prace nad kompleksowym opracowaniem technologii mycia w wodnych roztworach środków powierzchniowo-czynnych.

W wyniku podjętych badań opracowano i wdrożono do produkcji w maju 1983 r. proces mycia pakietów i podzespołów elektronicznych po lutowaniu na dwóch wydziałach - na wydziale montażu kalkulatorów elektronicznych oraz na wydziale montażu aparatury fizykochemicznej; natomiast w październiku 1983 r. na wydziale montażu pakietów komputerów i urządzeń komputerowych. Wdrożenie procesu mycia w wodnych roztworach środków powierzchniowo-czynnych przeprowadzono przy współpracy z Przedsiębiorstwem Wdrożenia i Upowszechnienia Postępu Technicznego i Organizacyjnego POSTEOR we Wrocławiu.

Charakterystykę materiałów stosowanych w procesie lutowania i mycia oraz sposób mycia w wodnych roztworach środków powierzchniowo-czynnych zawarto w dalszej części artykułu.

Charakterystyka stosowanych materiałów

● Topnik do lutowania

Zastosowany w procesie automatycznego lutowania topnik jest alkoholowym roztworem gliceryny lub/i glikolu polietylenowego o średnim ciężarze cząsteczkowym 1500, oraz aktywatorów zawierających między innymi kwasy organiczne. Na lutowane powierzchnie może być nanoszony w postaci piany lub metodą zanurzeniową. Może być stosowany zarówno do lutowania płytek obwodów drukowanych bez metalizowanych otworów jak i o metalizowanych otworach. Receptura stosowanego topnika objęta jest patentem 125256. Parametry fizykochemiczne przedstawiają się następująco:

● Olej lutowniczy

Opracowano dwa oleje lutownicze. Olej lutowniczy o nazwie Lutol przeznaczony jest do automatycznego lutowania na stojącej fali z iniekcją oleju do fali, olej lutowniczy o nazwie Lutoxol przeznaczony jest do lutowania automatycznego na stojącej fali spoiwa cynowo-ołowiowego z natryskiem oleju na falę. Obydwa oleje lutownicze mogą być stosowane do zabezpieczania spoiwa cynowo-ołowiowego w innych typach agregatów lutowniczych, w agregatach do wstępnego pokrywania wyprowadzeń elementów spoiwem cynowo-ołowiowym i do przetapiania metodą zanurzeniową spoiwa cynowo-ołowiowego na płytkach obwodów drukowanych o metalizowanych otworach.

Opracowane oleje lutownicze są całkowicie rozpuszczalne w wodzie i nie wywołują korozji metali, nie pienią się i są stabilne w zakresie temperatur stosowanych przy lutowaniu elementów i podzespołów elektronicznych. Brak

właściwości toksycznych, słaby zapach oraz wysoka temperatura zapłonu, nie mniej niż 250°C, oznaczona kubkiem otwartym wg PN-82/C-04008, czynią je bezpiecznym w użyciu. Zaś zdolność do biologicznej degradacji umożliwia odprowadzenie do ścieków pozostałości oleju lutowniczego usuwanych z pakietów w procesie lutowania. Olej lutowniczy Lutol jest ciałem stałym o temperaturze krzepnięcia 50°C, a olej lutowniczy Lutoxol w normalnych warunkach jest cieczą.

Ze względu na przygotowywanie w przedsiębiorstwie zgłoszenia patentowego do Urzędu Patentowego PRL informacje na temat składu chemicznego nie mogą być publikowane.

Oleje lutownicze opracowane zostały przy współpracy z Przedsiębiorstwem Wdrożenia i Upowszechnienia Postępu Technicznego i Organizacyjnego POSTEOR we Wrocławiu. Obecnie w ZE ELWRO trwają próby mające na celu wydłużenie czasu ich eksploatacji. Koszt materiałów do sporządzenia 1 kg oleju lutowniczego wynosi 100 zł.

● Mieszanka myjąca

Mieszanka myjąca do mycia pakietów i podzespołów elektronicznych objęta jest zgłoszeniem patentowym P241221. Mieszanka myjąca stanowi wodny roztwór środka powierzchniowo-czynnego, etanoloaminy i inhibitora korozji spoiwa cynowo-ołowiowego. Może ona być przygotowana bezpośrednio z handlowych składników lub przez rozcieńczenie uprzednio przygotowanego koncentratu. Stanowi dobry środek do usuwania z płytek drukowanych i podzespołów elektronicznych pozostałości oleju lutowniczego, topników lutowniczych oraz produktów ich rozkładu. Usuwa też skutecznie pozostałości topników kalafoniowych, stosowanych do międzyoperacyjnego zabezpieczenia obwodów bez metalizowanych otworów. Nie jest szkodliwa dla elementów elektronicznych tworzyw sztucznych, nie powoduje uszkodzeń na elementach i podzespołach elektronicznych.

Mycie pakietów i podzespołów elektronicznych może być prowadzone metodą zanurzeniową lub natryskiem. W przypadku zastosowania natrysku należy zastosować odpowiednie środki antypienne. Wypłukanie podzespołów po myciu w wodzie demineralizowanej zapewnia wysoką jakość procesu mycia. Mieszanka myjąca może być również stosowana do mycia elementów po wstępnym pokrywaniu wyprowadzeń elementów spoiwem cynowo-ołowiowym oraz do mycia płytek obwodów drukowanych po przetopieniu spoiwa cynowo-ołowiowego metodą zanurzeniową.

Proces technologiczny mycia w wodnych roztworach środków powierzchniowo-czynnych

Proces mycia w wodnych roztworach środków powierzchniowo-czynnych może być zrealizowany wg jednego z niżej podanych schematów:

I Schemat mycia

I Komora - Mycie zanurzeniowe w mieszaninie myjącej:

temperatura mycia - 40 - 50°C

czas mycia -

czas obciekania roztworu - 10 s.

II Komora - Mycie zanurzeniowe w wodzie wodociągowej:

temperatura mycia - 40 - 50°C

czas mycia -

czas obciekania - 10 s

częstotliwość wymiany wody - codziennie

III Komora - Mycie zanurzeniowe w wodzie demineralizowanej:

temperatura mycia -

czas obciekania -

czas mycia - 10 s

częstotliwość wymiany wody -

IV Komora - Mycie zanurzeniowe w wodzie demineralizowanej:

temperatura mycia -

czas mycia -

czas obciekania - 10 s

woda demineralizowana w komorze IV podlega wymianie po osiągnięciu przewodnictwa 50 $\mu\text{S}/\text{cm}$

V Suszenie - nadmuchi ciepłym powietrzem:

temperatura suszenia -

czas suszenia -

II Schemat mycia

I Komora - Mycie natryskowe bieżącą wodą:

temperatura mycia -

czas mycia -

czas obciekania -

II Komora - Mycie zanurzeniowe w mieszaninie myjącej:

temperatura mycia -

czas mycia -

czas obciekania -

III Komora - Mycie zanurzeniowe w wodzie wodociągowej:

temperatura mycia -

czas mycia -

czas obciekania -

IV Komora - Mycie natryskowe wodą demineralizowaną:

/po płukaniu w komorze V/

temperatura mycia -

czas mycia -

czas obciekania -

V Komora - Mycie natryskowe wodą demineralizowaną:

temperatura mycia -

czas mycia -

czas obciekania -

woda demineralizowana do płukania posiada przewodnictwo na poziomie 15 $\mu\text{S}/\text{cm}$; po przekroczeniu 50 $\mu\text{S}/\text{cm}$ ulega wymianie.

VI Suszenie - Nadmuchi ciepłym powietrzem:

temperatura suszenia -

czas suszenia -

W ZE ELWRO realizowany jest proces mycia pakietów wg obu schematów. Pakiety kalkulatorów elektronicznych, fizyko-chemicznej aparatury kontrolno-pomiarowej i automatyki podlegają procesowi mycia wg schematu I, a pakiety komputerów i urządzeń komputerowych wg schematu II. Procesy mycia prowadzi się w zmodernizowanych urządzeniach, w których uprzednio prowadzono proces mycia w roztworach opartych na freonie 113. Mycie wg schematu II zapewnia wyższą jakość w porównaniu z myciem wg schematu I, jednak jego techniczna realizacja napotyka na większe trudności.

Przybliżone wskaźniki techniczne zużycia materiałów w procesie mycia

Średnie zużycie roboczej mieszaniny myjącej do mycia pakietów po lutowaniu wynosi 2 dm^3/m^2 gabarytu konstrukcyjnego pakietów. Zaś zużycie wody demineralizowanej wynosi 30 dm^3/m^2 gabarytu konstrukcyjnego.

Ocena efektów ekonomicznych

W wyniku wprowadzenia technologii mycia pakietów po lutowaniu w wodnych roztworach środków powierzchniowo-czynnych i wynikających z tej technologii zmian w procesie lutowania uzyskano wielorakie korzyści. Do najcenniejszych z nich należą:

- poprawa jakości połączeń lutowanych wyrażająca się zmniejszoną ilością poprawek lutowniczych po lutowaniu automatycznym,
- wyeliminowanie materiałów do lutowania i mycia po lutowaniu importowanych z II obszaru płatniczego. Roczne oszczędności z tego tytułu wynoszą w ZE ELWRO 60000 zł .

- obniżenie kosztów mycia pakietów po lutowaniu.

Sprawdzony w ZE ELWRO proces mycia pakietów i podzespołów elektronicznych po lutowaniu w wodnych roztworach środków powierzchniowo-czynnych w wyniku prawie jednorocznej eksploatacji należy uznać za istotny postęp technologiczny w polskim przemyśle elektronicznym.

mgr inż. JERZY MAKOWIECKI
Zakład Przetwarzania Informacji
Tekstowej Instytutu Organizacji
Przemysłu Maszynowego

"INNOWACJE"

SYSTEM KONWERSACYJNEGO DOSTĘPU DO BANKU DANYCH O OPROGRAMOWANIU I UPOWSZECHNIENIU OPROGRAMOWANIA

W ramach prowadzonych w Instytucie Organizacji Przemysłu Maszynowego prac nad upowszechnieniem nowych rozwiązań powstał system komputerowego wspomaganie upowszechniania rozwiązań innowacyjnych. Jest to konwersacyjny wielodostępny system wyszukiwania informacji tekstowej. Cechy funkcjonalne i organizacja systemu czynią go uniwersalnym, może on służyć m. in. jednostkom administracji państwowej, przedsiębiorstwom, bibliotekom lub jednostkom usług informacyjnych. Oprogramowanie nie narzuca żadnych ograniczeń treści przechowywanych i przetwarzanych informacji. Informacje przechowywane są w bazach. Jedna baza dotyczy jednej dziedziny. System może obsługiwać do 15 baz równocześnie. W pierwszej wersji założona została jedna baza - informacji o innowacjach.

Obecnie powstaje kilka następnych baz do innych celów. Informacje w każdej bazie podzielone są na jednostki nazywane dokumentami. Dokument jest podstawową grupą informacji przetwarzanych równocześnie. Jeden dokument opisuje część lub całość jednego rozwiązania. Jedynym ogranicznikiem ilości dokumentów w bazie jest pojemność dostępnej pamięci dyskowej. Dokumenty podzielone są na pola w sposób ustalony dla każdej bazy. Wszystkie pola mają swoje nazwy. Dla dokumentów każdej bazy można określić najwyżej 32 pola. Dokumenty mogą zawierać wszystkie pola, lub tylko niektóre z nich. Wielkości pól są ograniczone: największe pole może zawierać do ok. 6000 znaków, przy czym przechowywana jest tylko część pola zawierająca dane. Znaczenie niektórych pól jest stałe dla wszystkich baz. Wyróżnione są m. in. pola: Numer Systemowy, Tytuł lub Nazwa, Autor itp. Informacje zapisywane są otwartym tekstem, podzielonym na zdania /kończone kropką/ i słowa /ciągi znaków różnych od spacji rozdzielone spacjami./. Należy za-

uważyć, że słowem jest każdy ciąg znaków niezależnie od jego znaczenia. Na etapie tworzenia bazy można zabronić interpretowania ciągów znaków zaczynających się od cyfry i/lub zawierających znaki specjalne jako słowa, aby uniknąć wypełniania słowników nieznaczącymi tekstami.

Wyszukiwanie informacji odbywa się trybem konwersacyjnym. Użytkownik zgłasza identyfikator i hasło oraz dokonuje wyboru bazy. Od tego momentu zaczyna się wyszukiwanie. Jest to proces iteracyjny, umożliwiający uściślanie pytania w kolejnych fazach poprzez powołanie się na któreś z wcześniej zadanych pytań i uzupełnienie go o nowe kryteria. Użytkownik może wybierać dokumenty zawierające określone słowa lub ich nie zawierające. Może szukać grup słów w układzie dowolnym lub podanym przez niego, może określać czy dana grupa słów ma być odnaleziona w dokumencie, w dowolnym polu, w określonym polu, albo w jednym zdaniu określonego lub dowolnego pola. Może wreszcie podawać nie całe słowa, a tylko początkowe litery /np. jeśli obejmują one rdzeń rodziny słów interesujących użytkownika/. Pytanie ma postać ciągów słów oddzielonych tzw. operatorami /słowami kluczowymi opisującymi łączące je relacje/, podzielonych na grupy parami nawiasów. Operatory mają określone priorytety, zaś nawiasy służą, podobnie jak w wyrażeniach arytmetycznych, do wymuszenia innej kolejności interpretacji.

Przykład /uwaga: numer przed pytaniem nadawany jest automatycznie przez system/
001 komputer - oznacza żądanie wybrania wszystkich dokumentów zawierających słowo "komputer"

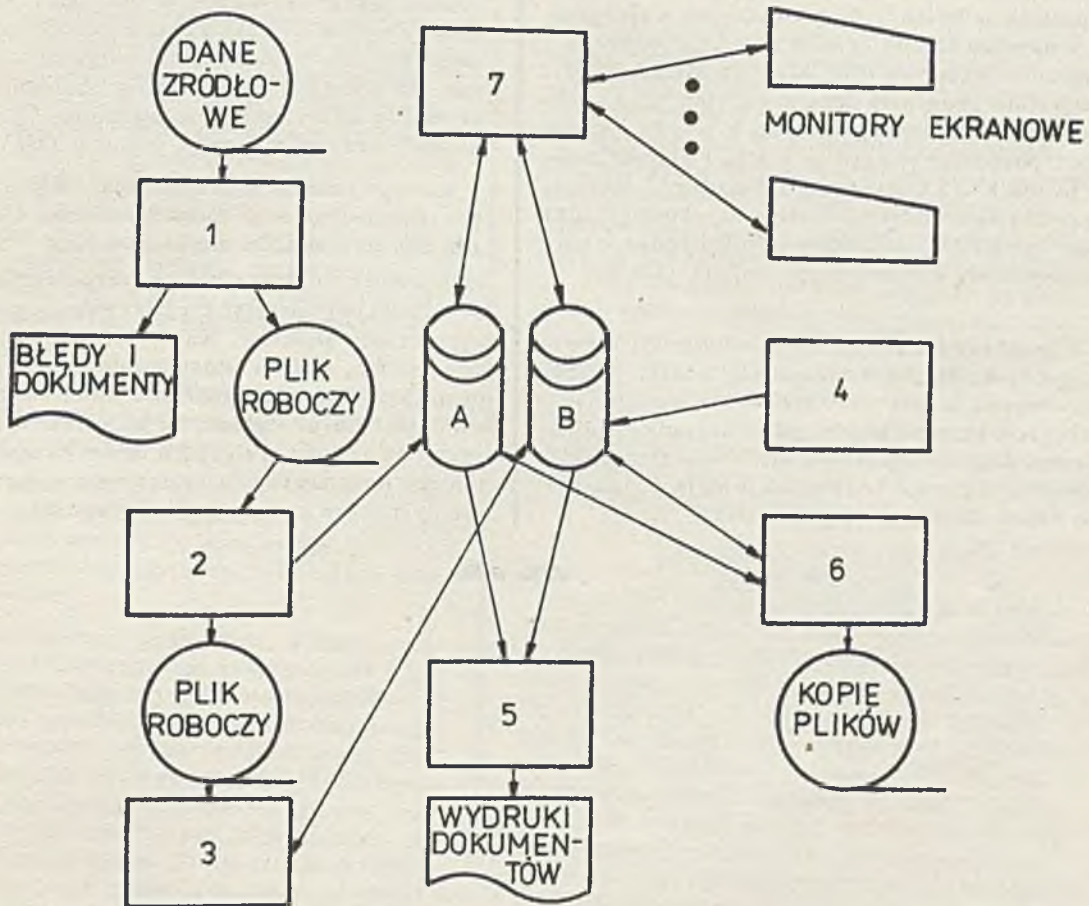
002 komput \$ - oznacza żądanie wybrania dokumentów zawierających słowa zaczynające się od "komput" /znak \$ oznacza dowolną końców-

kę/. Odpowiedzią będą wszystkie dokumenty zawierające słowa "komputer", "komputera", "komputerowy", "komputeryzacja".
 003 wyszukiwanie i /danych lub informacji/
 - oznacza żądanie wybrania dokumentów zawierających słowa "wyszukiwanie" i "danych" lub zawierających słowo "informacji"
 004 wyszukiwanie i /danych lub informacji/
 - oznacza żądanie wybrania dokumentów zawierających słowa "wyszukiwanie" i "danych" lub słowa "wyszukiwanie" i "informacji".

W dowolnym momencie użytkownik może życzyć sobie wyświetlenia dokumentów będących odpowiedzią na dowolne spośród zadanych przez niego pytań. Wyświetlane mogą być całe dokumenty lub tylko pola wskazane przez użytkownika. Użytkownik może też życzyć sobie wydrukowania aktualnie oglądanej odpowiedzi oraz wyświetlenia treści wszystkich pytań zadanych od początku sesji. System ma wbudowane me-

chanizmy ochrony danych przed nieupoważnionym dostępem. Obejmuje rejestr użytkowników, haseł i uprawnień. Ochrona dotyczy całych baz, określonych pól w dokumentach bazy lub konkretnych dokumentów. Dane, do których użytkownik nie ma dostępu nie są ani wyświetlane ani analizowane przy wyszukiwaniu.

System został zaprojektowany do wykorzystania na maszynie IBM 370 z systemem OS/VS1. W jego strukturze /rys.1/ można wyróżnić pięć grup programów realizujących:
 - zakładanie baz danych, weryfikacja danych /1/,
 - zakładanie plików baz /2/, aktualizacja plików opisu baz /3/,
 - zakładanie plików pomocniczych i roboczych systemu /4/,
 - generację biuletynów, indeksów i innych wydruków danych z bazy /5/,
 - konwersacyjną współpracę z użytkownikami /7/.



Rys.1. Struktura systemu "Innowacje". A. pliki bazy danych. B. pliki pomocnicze i robocze /rejestr użytkowników, opisy baz itp./. 1. programy wstępnej analizy danych. 2. programy tworzenia i rozszerzania plików bazy. 3. programy aktualizacji plików pomocniczych. 4. programy zakładania plików pomocniczych. 5. programy wydruku biuletynów, indeksów i wydawnictw nietypowych /tworzonych tylko dla jednej bazy/. 6. programy kopiowania i odtwarzania plików systemu. 7. programy konwersacyjnej współpracy z użytkownikami

- kopiowanie i ewentualne odtwarzanie plików systemu /6/.

Zakładanie baz danych odbywa się wsadowo w trzech etapach. Pierwszy to kontrola formalna danych i tworzenie pliku roboczego zawierającego dane zaakceptowane. Na podstawie wydruku błędów poprawia się dane wejściowe, następnie powtarza się kontrolę. Należy to czynić aż do uzyskania zadowalającej poprawności formalnej danych. Wówczas tworzy się wydruk pliku roboczego, przeprowadza sprawdzenie poprawności merytorycznej i poprawia dane wejściowe. Wszystkie te czynności należy wykonywać tak długo, aż dane osiągną zadowalającą poprawność.

Etap drugi najbardziej kosztowny, to tworzenie plików bazy. Można założyć bazę nową lub dołączyć nowe dokumenty do bazy istniejącej.

W następnym etapie aktualizacji podlegają pliki opisów baz /np. aktualizacja ilości dokumentów w bazie/. Konwersacyjną współpracę z użytkownikami system realizuje poprzez monitory ekranowe IBM 3277 lub MERA 7910. Wszystkie programy konwersacyjne oraz większość wsadowych napisane są w języku Assembler, pozostałe programy w PL/1 /kompilowane przez PL/1 Optimizing Compiler/.. Używaną początkowo metodą dostępu telekomunikacyjnego był BTAM zmieniony później /poprzez wymianę modułu komunikacyjnego/ na TCAM.

Wyszukiwanie odbywa się metodą list inwersyjnych. Każda baza składa się z kilku plików fizycznych, w tym pliku tekstów dokumentów, pliku list inwersyjnych, pliku słownika, pliku skorowidza dokumentów i in. Wszystkie pliki używane w czasie konwersacji mają organizację zapewniającą dostęp bezpośredni.

Do eksploatacji systemu potrzebne są:

- Maszyna IBM 370 z systemem OS/VS1.
- Strefa 384 K PAO na użytek systemu.
- Jednostka pamięci dyskowej 2314.
- 2 jednostki 9-ścieżkowej pamięci taśmowej /przy zakładaniu baz/.
- Monitor ekranowy IBM 3277 lub MERA 7910.

Zalecane jest użycie pamięci dyskowej 3330 i wielu monitorów ekranowych. System został uruchomiony i przedstawiony na maszynie IBM/370, model 145, w Centrum Obliczeniowym Ministerstwa Hutnictwa i Przemysłu Maszynowego. Przeprowadzono testy czasu odpowiedzi systemu oraz zmian kosztów eksploatacji przy wzrastającej wielkości bazy. Testy przeprowadzono na bazach od 200 do 1000 dokumentów. Czas odpowiedzi na pojedyncze pytanie wahał się od 1 do 8 sekund z tendencją do wzrostu wraz z wielkością bazy oraz komplikacją kryteriów w pytaniu. Nie zanotowano istotnych zmian czasu reakcji systemu przy zwiększaniu liczby użytkowników równocześnie aktywnych w systemie. Bardzo istotny wpływ na czas reakcji ma obciążenie maszyny przez inne zadania aktywne w systemie. Dostrzegalna jest również zależność od używanej metody teleprzetwarzania /BTAM/ TCAM/ oraz parametrów systemu OS/VS1.

Koszt wyszukiwania mierzono ilością dostępuów dyskowych oraz zużytym czasem CPU. Dla obu parametrów zaobserwowano wzrost słabszy niż liniowy wraz ze wzrostem bazy, co odpowiada teoretycznej krzywej logarymicznej wzrostu kosztów. Koszt zainicjowania pracy systemu, duży w stosunku do kosztu jednego pytania, wzrasta nieznacznie wraz ze wzrostem wielkości bazy. Zaleca się więc tworzenie dużych baz /rzędu tysięcy, a nawet dziesiątków tysięcy dokumentów/ i intensywne wykorzystanie systemu w czasie jego aktywności.

mgr inż. ANDRZEJ ADEREK

Przemysłowy Instytut
Automatyki i Pomiarów

OPROGRAMOWANIE ZDECENTRALIZOWANEGO SYSTEMU AUTOMATYKI KOMPLEKSOWEJ "MIR-PROWAY"

Zdecentralizowany układ automatyki kompleksowej w systemie MIR-PROWAY składa się z pewnej liczby stacji dołączonych do linii transmisji danych. Stacja jest to dowolne urządzenie automatyki, które może wymieniać ramki informacji z innymi stacjami, zgodnie z protokołami określonymi w standardzie PROWAY [1]. Stacją może być minikomputer zbierający dane z procesu, sterownik DDC, konsola operatora procesu, wyświetlacz, czytnik etykiet, itp. /rys. 1/. W jednym układzie automatyki może znajdować się maks. 100 stacji.

Oprogramowanie niezbędne do wdrożenia układu automatyki opartego o urządzenie PROWAY obejmuje dwie części:

- oprogramowanie podstawowe, które organizuje pracę stacji i zarządza komunikacją między stacjami zgodnie ze standardem PROWAY,
- oprogramowanie użytkowe realizujące zadania automatyki.

Oprogramowanie podstawowe w małym stopniu zależy od konkretnego zastosowania i jest uniwersalne. Jego przygotowanie przekracza możliwości przeciętnego użytkownika; z tego względu powinno być dostarczane przez producenta sprzętu lub wyspecjalizowane przedsiębiorstwo. Oprogramowanie użytkowe będzie natomiast wykonywane indywidualnie dla każdego zastosowania. Programistom użytkowym powinny być jednak udostępnione odpowiednie narzędzia programowe, takie jak biblioteki podprogramów oraz języki wyższego rzędu i zorientowane problemowo, uniwersalne systemy programowania.

W niniejszym artykule przedstawiono prace, jakie prowadzone są w Przemysłowym Instytucie Automatyki i Pomiarów w zakresie oprogramowania dla systemu MIR-PROWAY. Oprogramowanie to przygotowywane jest dla pod-

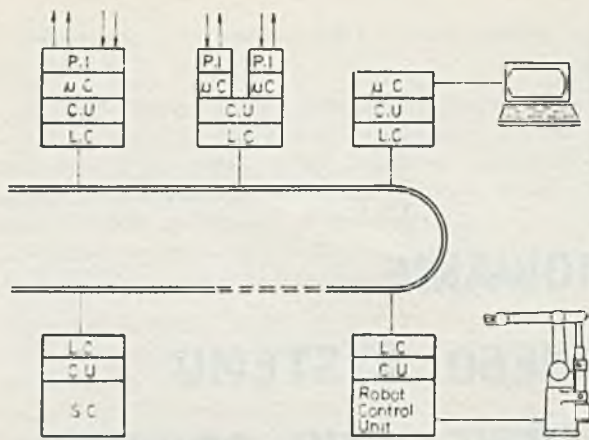
stawowej grupy procesorów przewidywanych do stosowania w systemie MIR-PROWAY. Są to mikrokomputery MM80, MM86 oraz minikomputery SM. Przedstawione oprogramowanie kontrolera komunikacyjnego, biblioteka programów do współpracy z tym kontrolerem oraz uniwersalne systemy operacyjne czasu rzeczywistego umożliwiają tworzenie użytkowego oprogramowania dla układów automatyki. Programowanie na tym poziomie wymaga jednak specjalizowanego przygotowania z zakresu informatyki, dlatego zaleca się do stosowania uniwersalny system programowania SZPAK-DS przedstawiony w dalszej części artykułu. W ostatnim rozdziale artykułu wyspecyfikowane zostaną narzędzia programowe, jakie przewiduje się udostępnić programistom systemów MIR-PROWAY.

Oprogramowanie kontrolera komunikacyjnego

Kontroler komunikacyjny wchodzi w skład każdej stacji MIR-PROWAY i łączy procesory użytkownika z linią transmisji poprzez sterownik linii. Oprogramowanie kontrolera komunikacyjnego realizuje protokoły ścieżki i magistrali określone w standardzie PROWAY oraz protokoły sieci. Usytuowanie logiczne kontrolera komunikacyjnego w stacji ilustruje rys. 2. Przedstawiono na nim podstawowe funkcje realizowane przez oprogramowanie kontrolera komunikacyjnego. Oprogramowanie to będzie w całości dostarczane przez producenta mikroprocesorowych urządzeń MIR-PROWAY.

Podprogramy do współpracy z kontrolerem komunikacyjnym PROWAY

Dla programowania przez użytkownika urządzeń automatyki połączonych magistralą PROWAY zaleca się stosowanie uniwersalnych systemów programowania wyższego rzędu. W sy-



Rys. 1. Przykład układu automatyki w systemie MIR-PROWAY: L. C. - sterownik linii, P. I. - łączące z obiektem, C. U. - kontroler komunikacyjny. S. C. - komputer nadrzędny

stemach tych wszystkie czynności związane z obsługą magistrali PROWAY będą wykonywane przez standardowe oprogramowanie firmowe. Oprogramowanie firmowe będzie realizowało m. in. takie czynności jak:

- sprowadzanie i wysyłanie danych z/do stacji zdalnych w oparciu o symboliczną /logiczną/ nazwę,

- aktualizację lokalnej bazy danych stacji w zakresie danych rejestrowanych w innych stacjach,
- wyprowadzanie komunikatów o alarmach,
- przeładowywanie oprogramowania stacji po restarcie itd.

Oprogramowanie firmowe będzie również wykonywało funkcje zarządzania odpowiadające poziomowi protokołów sieci. W części zastosowań użytkownik będzie wołał samodzielnie zaprogramować ww. czynności lub będzie chciał wykonać specjalizowany system programowania zorientowany na wąską grupę zastosowań. W tym celu przewiduje się opracowanie biblioteki podprogramów umożliwiających programowanie elementarnych funkcji PROWAY. Będą w niej uwzględnione funkcje dopuszczone do stosowania przez użytkownika w standardzie PROWAY. Biblioteka będzie zrealizowana w formie wielowejściowej. Przewiduje się opracowanie takiej biblioteki dla systemu operacyjnego RT mikrokomputera MM80, dla systemu operacyjnego RT mikrokomputera MM86 oraz dla systemu operacyjnego RSX-11 minikomputerów SM. Systemy te będą również posiadały mechanizmy aktywowania odpowiednich zadań użytkownika w związku ze zdarzeniami w PROWAY. Poniżej przedstawiono w zarysie formę wywołań podprogramów biblioteki do współpracy z kontrolerem komunikacyjnym PROWAY.

1. 1. Podprogram wysyłania danych z potwierdzeniem odbioru

Postać wywołania: CAL SENDA /du, n, sa, m/ gdzie:

du^{1/} - adres stacji przeznaczenia, do której mają być przesłane dane, $1 \leq da \leq 127$

/1 bajt/

n - liczba bajtów danych, $1 \leq n \leq 1024$

/2 bajty/

sa^{2/} - adres początku danych do przesłania

/3 bajty/

m - informacja o wykonaniu podprogramu

/1 bajt/

$m \geq 0$ oznacza, że wykonanie podprogramu nie zostało zakończone,

bit 7m = 1 oznacza zakończenie obsługi podprogramu; ewentualne błędy wykonania sygnalizowane są na bitach 0-4,

bit 0 = 1 oznacza błąd transmisji /np. przerwanie transmisji z powodu przekroczenia dopuszczalnego czasu lub brak odpowiedzi stacji przeznaczenia mimo wykonania 7 prób transmisji/,

bit 1 = 1 stacja wykonująca podprogram /lokalna/ odłączona od linii,

bit 2 = 1 stacja lokalna odłączona od linii z powodu przekroczenia czasu transmisji,

bit 3 = 1 stacja przeznaczenia jest niedostępna /np. brak wolnego bufora na dane odbierane/

bit 4 = 1 niedozwolony parametr wywołania podprogramu.

1. 2. Podprogram globalnego wysyłania danych

Postać wywołania: CAL SENDG /n, sa, m/ gdzie:

n, sa - patrz pkt. 1. 1.

m - informacja o wykonaniu podprogramu

$m \geq 0$ oznacza, że wykonywanie podprogramu nie zostało zakończone,

$m < 0$ oznacza zakończenie wykonywania podprogramu; ewentualne błędy wykonania stwierdzone w stacji lokalnej są sygnalizowane następująco:

bit 1 = 1 oznacza odłączenie stacji lokalnej od linii,

bit 2 = 1 oznacza odłączenie stacji lokalnej od linii z powodu przekroczenia czasu transmisji,

bit 4 = 1 oznacza niedozwolony parametr wywołania podprogramu.

U w a g a:

Po wykonaniu podprogramu nie jest dostarczana do stacji lokalnej informacja o odebraniu danych przez stację zdalną.

1. 3. Podprogram zgłaszania żądania dostarczenia danych

Postać wywołania: CAL REQUEST /du, n, sa, m/

1/ du - destination unit

2/ sa - start address

gdzie:

du - adres stacji, z której użytkownik żąda dane

n, sa - liczba bajtów i adres początku obszaru pamięci, do którego mają być wpisane żądane dane

m - patrz pkt. 1.1.

1.4. Podprogram zgłoszenia gotowości danych do wysłania na żądanie stacji zdalnej

Postać wywołania: CAL READYR /du, n, sa, 1/

gdzie:

du - adres stacji przeznaczenia /żądającej/

n, sa - patrz pkt. 1.1.

l - określa stan wykonania podprogramu i dostępności danych określonych przez n, sa.

l = 1 oznacza, że dane wykorzystywane są przez PROWAY. Stan ten trwa od odebrania żądania do pojawienia się w linii nowej ramki typu "baiton". Gdy l = 1, to dane określone przez n, sa są niedostępne dla użytkownika.

l = 0 oznacza zakończenie obsługi podprogramu - patrz opis m w pkt. 1.1.

Dane określone przez parametry n, sa wysyłane są do stacji du, gdy zostanie odebrane żądanie z tej stacji.

U w a g a:

Podane w du wartości ϕ oznaczają, że dane w sa są dostępne dla wszystkich stacji, które mogą być stacjami typu iniciator.

1.5. Podprogram kasowania zgłoszenia gotowości danych do wysłania na żądanie

Postać wywołania: CAL DELR /du, m/

Podprogram kasuje wcześniej zadeklarowaną gotowość danych do wysłania na żądanie zgłoszonych podprogramem READYR, przeznaczonych dla stacji du.

gdzie:

du - patrz pkt. 1.4.

m - informacja o wykonaniu podprogramu

m < 0 oznacza, że podprogram został wykonany; ewentualne błędy podane są na pozycjach 4 i 5,

bit 4 = 1 oznacza błąd parametrów /brak w liście zgłoszeń pozycji do stacji du/

bit 5 = 1 oznacza, że pozycja jest zajęta przez PROWAY i nie została skasowana.

1.6. Podprogram zgłoszenia obszaru pamięci na dane wejściowe z PROWAY

Postać wywołania: CAL INBUF /n, ba, 1, m/

gdzie:

n - liczba bajtów bufora

ba - adres początku bufora

l - wskaźnik stanu bufora

l = 0 oznacza, że bufor jest wolny

l = 1 oznacza, że bufor jest zajęty przez PROWAY.

l < 0 oznacza, że w buforze znajdują się gotowe dane do wykorzystania przez użytkownika. Bufor jest niedostępny dla PROWAY.

Podprogram ten powinien być wykonany w każdej stacji przynajmniej jeden raz w czasie inicjacji.

m - wskaźnik wykonania. Po wykonaniu podprogramu w m znajduje się informacja o błędzie. Bit 4 = 1 oznacza błąd parametru.

1.7. Podprogram pobierania listy aktywnych stacji

Postać wywołania: CAL GIVELL /sa, m/

gdzie:

sa - adres początku obszaru 17 bajtów.

Po wykonaniu podprogramu w sa znajdują się następujące dane:

bajt pierwszy: = 0 oznacza, że nie było zmiany stanu żadnej stacji od poprzedniego wykonania podprogramu GIVELL

= 1 - była przynajmniej jedna zmiana.

bajty 2 - 17: na poszczególnych bitach podany jest stan stacji.

Numer kolejny bitu odpowiada adresowi stacji.

Wartość bitu 0 oznacza stację nieaktywną,

wartość 1 - aktywną.

m - patrz pkt. 1.6.

1.8. Podprogram ustawiania ograniczenia długości obszaru danych dla programów SENDA, SENDG

Postać wywołania: CAL SETNMAX /n, m/

gdzie:

n - maksymalna dopuszczalna długość obszaru danych do wysłania dla podprogramów SENDA, SENDG, $1 \leq n \leq 1024$

m - wskaźnik wykonania, m = 0 oznacza instrukcję odrzuconą /nie dozwolony parametr/.

1.9. Podprogram zmiany stanu dołączenia stacji do linii

Postać wywołania: CAL SETLCS /s, m/

gdzie:

s - żądany stan dołączenia linii

s = 0 oznacza dołącz stację do linii

s = 1 oznacza odłącz stację od linii

m - wskaźnik wykonania, m = 0 oznacza efektywne wykonanie, m = 1 oznacza instrukcję nieefektywną /nie spowodowała zmiany stanu/.

1.10. Podprogram pobierania licznika błędów

Postać wywołania: CAL GIVERC /c, m/

gdzie:

c - po wykonaniu podprogramu zawiera liczbę nieudanych prób transmisji /modulo 65536/

m - patrz pkt. 1.6.

1.11. Podprogram pobierania stanu dołączenia stacji do linii

Postać wywołania: CAL GIVELCS /s, m/

gdzie:

pierwszy bajt s = 0 oznacza, że nie było zmiany stanu dołączenia linii od poprzedniego odczytu,

= 1 - była zmiana.

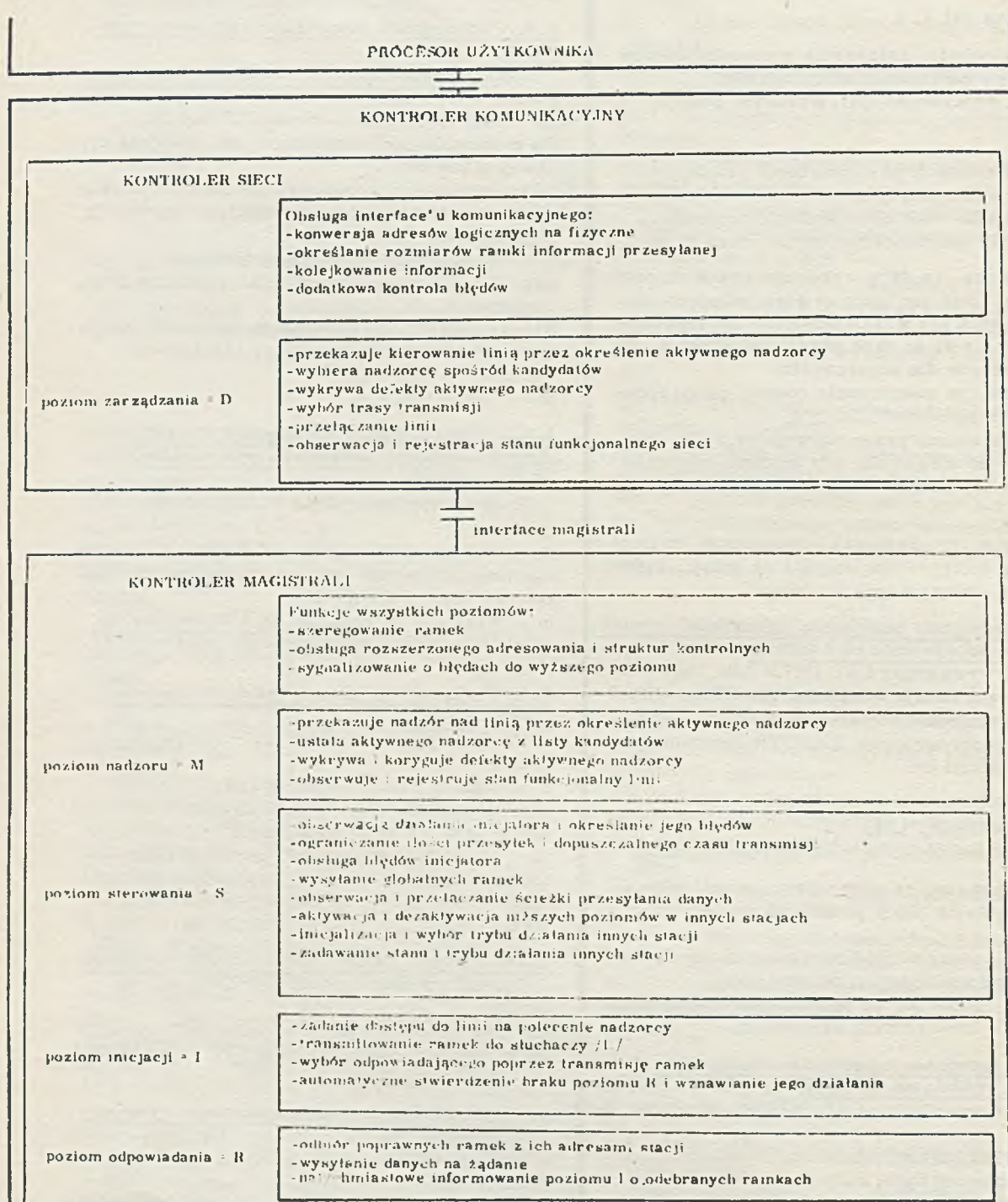
drugi bajt s - oznacza stan dołączenia linii

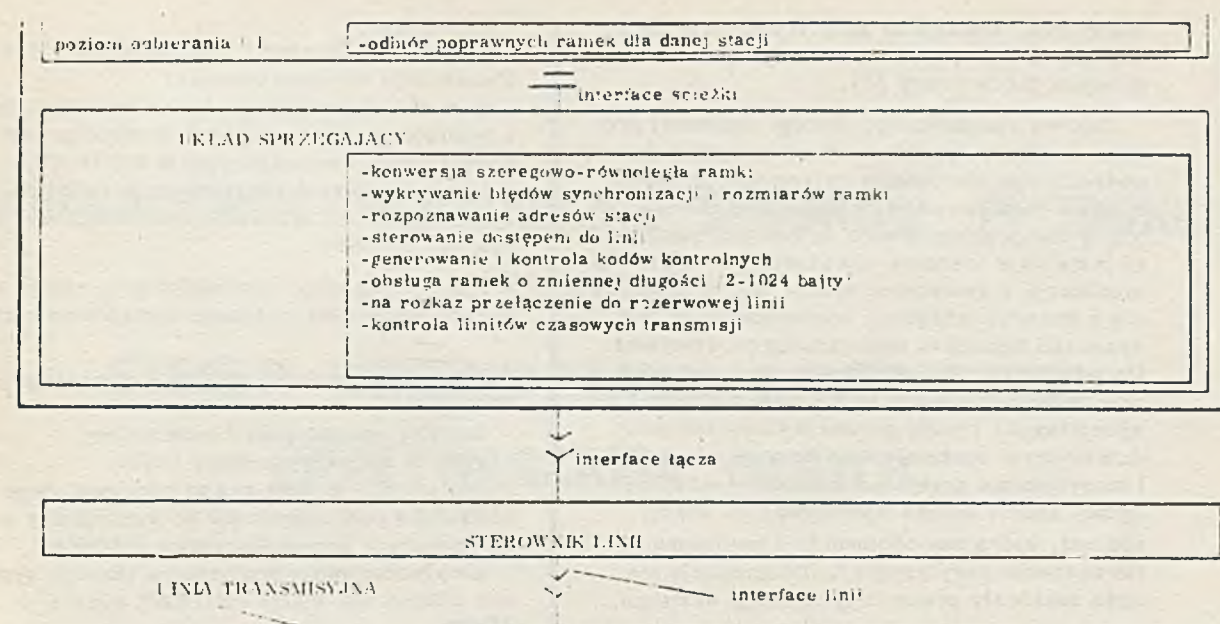
s = 0 - stacja jest dołączona do linii

s = 1 - odłączona.

m - patrz pkt. 1.6.

Rys. 2. Położenie /logiczne/ kontrolera komunikacyjnego w strukturze Stacji PROWAY





1.12. Podprogram restartu stacji zdalnej

Postać wywołania: CAL RECOVERY /du,

c, m/

gdzie:

du - adres stacji zdalnej

c - zawiera numer rozkazu $c = 1 \vee 2 \vee 3$

m - patrz pkt. 1.1.

System programowania SZPAK DS dla układów automatyki rozłożonych przestrzennie

Przygotowanie oprogramowania użytkowego dla rozłożonego przestrzennie układu automatyki jest zadaniem znacznie trudniejszym niż w przypadku układu z jednym centralnym komputerem. Oprócz zaprogramowania zadań lokalnych dla wielu, często różnych typów, procesorów użytkownika dodatkowo należy zorganizować efektywny i niezawodny dostęp do danych rozproszonych w różnych stacjach oraz zapewnić efektywne wykorzystanie środków transmisji PROWAY.

Przedstawione w poprzednim rozdziale podprogramy umożliwiają zakodowanie wymienionych funkcji. Wymagają jednak od użytkownika wysokiej znajomości m. in. zagadnień systemów operacyjnych czasu rzeczywistego, organizacji transmisji wewnątrz sieci i zagadnień baz danych czasu rzeczywistego. Przewiduje się, że oprogramowanie użytkowe będzie przygotowywane bezpośrednio w oparciu o te podprogramy tylko dla niewielkich /z małą liczbą stacji/ nietypowych zastosowań.

Systemem programowania zalecanym do stosowania dla układów automatyki opartych o PROWAY będzie SZPAK DS. Będzie to zmodernizowana wersja systemów SZPAK i SZPAK

77 zrealizowanych dla zestawów sprzętu ODRA -1325-SMA i MERA-400-PI, sprawdzonych w wielu zastosowaniach przemysłowych. Użycie tego systemu uwolni programistę od wielu problemów związanych z organizowaniem pracy stacji, organizowaniem bazy danych oraz rejestracją danych z procesu; całkowicie uwolni również od zagadnień transmisji danych w systemie PROWAY. Posługiwanie się tym systemem będzie proste i nie wymagające specjalizowanego wykształcenia informatycznego. Użytkownik przygotowuje oprogramowanie w systemie SZPAK, posługując się pojęciami automatyki takimi jak: zmienna procesu, stany, stała całkowania regulacji PID itd. Fakt, że układ automatyki jest rozłożony przestrzennie i ma wiele procesorów, będzie praktycznie bez większego znaczenia dla programisty. Definicja systemu będzie w dużym stopniu zgodna z opisem systemu SZPAK-77 podanym w pracy [2].

Dane w systemie będą identyfikowane w oparciu o nazwę zmiennej procesu. Lokalnie w ramach stacji, do której jest przywiązana zmienna procesu /ZP/ nazwy będą 4-znakowe /litera i 3 cyfry/. Nazwa globalna będzie składała się z numeru stacji i nazwy lokalnej. Nazwy globalne umożliwią sięganie do danych z baz danych stacji zdalnych. Dane rejestrowane z procesu, które będą używane tylko w ramach stacji lokalnych oraz dane zmieniane sporadycznie /np. ograniczenia/ będą przechowywane w bazie danych stacji macierzystej dla danej ZP. Dane aktualizowane okresowo będą mogły być przechowywane w stacji macierzystej oraz w stacjach zdalnych, które wykorzystują te dane. Aktualizacja oryginału i duplikatów będzie odbywała się auto-

matycznie. Dostęp do danych będzie programowany z wykorzystaniem podprogramów podobnych jak w pracy [2].

Zakres standardowej obsługi zmiennej procesu zostanie rozszerzony o algorytmy do bezpośredniego sterowania cyfrowego. Zmianie ulegnie rola obwodów regulacji cyfrowej. Zostaną one przystosowane do nowych możliwości jakie daje technika urządzeń PROWAY. Komunikacja z systemem będzie mogła odbywać się z różnych urządzeń komunikacji, w tym urządzeń będących samodzielnymi stacjami. Dodawanie nowych zmiennych procesu oraz programów użytkowych /uzupełniających i specjalnych/ będzie można wykonywać przy działającym systemie. Ładowanie programów i wczytywanie zmiennych do danej stacji lokalnej będzie można wykonywać ze stacji zdalnej, która ma odpowiednią konfigurację /urządzenia peryferyjne/. Te operacje nie będą zakłócały pracy innych stacji systemu.

Narzędzia programowe do przygotowywania oprogramowania dla układów automatyki w systemie MIR-PROWAY

Oprogramowanie jakie będzie dostępne dla programistów urządzeń mikroprocesorowych MIR-PROWAY obejmie narzędzia programowe do przygotowywania oprogramowania t. j. assembly, assembly skrócone, symulatory, programy monitora oraz systemy operacyjne i pakiety oprogramowania do pracy w czasie rzeczywistym. Oprogramowanie skrócone jest przygotowywane dla komputera SM-4. Pozostałe oprogramowanie będzie dostępne na mikrokomputerach MM80, MM86 oraz na minikomputerze SM-4.

Obecnie dla mikrokomputera MM80 dostępne jest bogate oprogramowanie w tym:

- assembler,
 - assembler skrótny i symulator na SM-4.
- Przewiduje się opracowanie:
- systemu operacyjnego czasu rzeczywistego z biblioteką podprogramów do współpracy z kontrolerem komunikacyjnym PROWAY,
 - oprogramowania realizującego funkcje systemu SZPAK dla stacji MIR-PROWAY opartej o MM80.

Dla mikrokomputera MM86 przewiduje się opracowanie następującego oprogramowania:

- assembler skrótny ASM86 i symulator na SM-4,
- monitor operatorski i assembler,
- system operacyjny klasy UNIX,
- system operacyjny czasu rzeczywistego z biblioteką podprogramów do współpracy z kontrolerem komunikacyjnym PROWAY,
- oprogramowanie realizujące funkcje systemu SZPAK dla stacji PROWAY opartej o MM86.

Dla minikomputera SM-4 przewiduje się opracowanie następującego oprogramowania do pracy w czasie rzeczywistym:

- biblioteka podprogramów do współpracy z kontrolerem komunikacyjnym PROWAY dla systemu operacyjnego RSX-11M,
- biblioteka podprogramów dostępu do rozproszonej bazy danych systemu SZPAK-DS.

L i t e r a t u r a :

- [1] Process data highway /PROWAY/ for distributed process control systems IEC TC55, Draft Document 65A /Central Office/ 10.
2. SZPAK-77: Definicja i zasady programowania. Podręcznik programowania PIAP.

NORMATYWNE PLANOWANIE ŚRODKÓW PRODUKCJI

SYNOPS

SYSTEM WPROWADZANIA I WSTĘPNEGO PRZETWARZANIA DANYCH

W ZETO-Katowice opracowany został system EPD "Normatywne Planowanie Środków Produkcji - SYNOPS" automatyzujący przetwarzanie danych podczas technicznego przygotowania produkcji i planowania produkcji w przedsiębiorstwie. SYNOPS może być eksploatowany na komputerach RIAD i IBM 360/370 pod nadzorem systemu operacyjnego OS. Jest on systemem powtarzalnym, co oznacza, że spełnia wymogi wielu użytkowników i może być wdrażany w przedsiębiorstwach różnych branż. Użytkownikami systemu SYNOPS mogą być przede wszystkim przedsiębiorstwa przemysłu maszynowego oraz takie przedsiębiorstwa innych branż, które posiadają dokumentację konstrukcyjno-technologiczną. Przedsiębiorstwa te mogą być jedno lub wielozakładowe. SYNOPS, w obecnym stadium opracowania, realizuje następujące czynności z zakresu tpp i planowania produkcji:

- zakładanie bazy normatywnej na podstawie dokumentacji konstrukcyjno-technologicznej,
- utrzymywanie w aktualnym stanie norm konstrukcyjno-technologicznych związanych z wyrobami finalnymi, zespołami, podzespołami i detalami występującymi w całym asortymencie produkcji przedsiębiorstwa,
- planowanie roczno-kwartalne i kwartalno-miesięczne produkcji w zakresie pracochłonności i materiałochłonności wyrobów,
- bilansowanie pracochłonności planu produkcji z możliwościami stanowisk pracy i ich obsad,
- emisja elementów dokumentacji produkcyjnej /dokumenty pobrania materiału RW, specyfikacje części na zlecenia,/,
- rozporządzanie wstępnej jednostkowej kalkulacji kosztów wyrobów.

Celem systemu jest usprawnienie zarządzania przedsiębiorstwem w zakresie:

a/ technicznego przygotowania produkcji poprzez:

- usprawnienia związane z obiegiem dokumentacji konstrukcyjno-technologicznej, które stanowią pomoc dla konstruktora i technologa poprzez wyeliminowanie wielu pracochłonnych prac,
- wyemitowanie zwinienia czyli specyfikacji zastosowania w wyrobach złożonych oraz zwinienia montażowego /rozdetalowania/ wyrobów złożonych, co w efekcie daje możliwość pokazania struktury wyrobu od wierzchołka do podstawy, tzn. od całości wyrobu, poprzez wszystkie zespoły i podzespoły, aż do wszystkich jego elementów podstawowych, to znaczy części i materiałów. Zestawienia te są bardzo przydatne konstruktorom i technologom w ich codziennej pracy
- wyeliminowanie części prac związanych z emisją dokumentacji produkcyjnej.

b/ planowania produkcji poprzez:

- ułatwienie oceny możliwości realizacji przyjętej wersji planu oraz usprawnienie budowy optymalnego planu produkcji,
- trafne wyznaczanie zadań dla poszczególnych komórek produkcyjnych z określeniem norm czasowych i materiałowych,
- ułożenie harmonogramu produkcji półfabrykatów z uwzględnieniem braków i zapasów tak, aby ich produkcja wyprzedzała zapotrzebowanie w harmonogramie produkcji wyrobów finalnych,
- wyemitowanie potrzeb materiałowych na wyroby przewidziane w planie produkcji, co stanowi cenną pomoc dla służb zaopatrzenia przy opracowaniu planów zakupów materiałów.

c/ planowania kosztów poprzez wyemitowanie wstępnej jednostkowej kalkulacji kosztów wyrobów finalnych i części zamiennych.

System SYNOPS funkcjonuje w oparciu o następujące zbiory podstawowe:

- kartotekę konstrukcyjno-technologiczną /KKT/,
- zbiór nazw i symboli stanowisk,
- zbiór parametrów stanowisk,
- zbiór stawek placowych,
- zbiór parametrów wydziałów,
- zbiór planu produkcji,
- zbiór półfabrykatów.

Ponadto system korzysta z kartoteki indeksu materiałowego utworzonej w systemie gospodarki materiałowej SYGMAT. Kartoteka konstrukcyjno-technologiczna /KKT/ zakładana jest w oparciu o następujące dokumenty:

- Wykaz części lub zespołów;
- Karta technologiczna;
- Jednostkowa norma zużycia materiału;
- Wykaz serii ekonomicznych i cykli produkcyjnych.

KKT zbudowana jest z pakietów informacji. Każdy pakiet stanowi odbicie konstrukcyjnej struktury oraz technologii wytwarzania jednego asortymentu. Przez asortyment rozumie się: wyrób finalny, zespół, podzespół, lub detal. Część konstrukcyjną pakietu stanowią informacje z dokumentu "Wykaz części lub zespołów" lub dokumentu "Jednostkowa norma zużycia materiału". Część technologiczną pakietu stanowią informacje z dokumentu "Karta technologiczna". Każdy pakiet zawiera też informacje ogólne o asortymencie, takie jak: nazwa asortymentu, numer rysunku, długość cyklu produkcyjnego, seria ekonomiczna. Jeśli jakaś część wchodzi w skład kilku asortymentów, zakładany jest dla niej tylko jeden pakiet w KKT. Każda pozycja części konstrukcyjnej pakietu KKT, jak również każda operacja technologiczna, może mieć kilka wersji różniących się datami początku ich ważności. KKT jest zbiorem sekwencyjnym.

Na podstawie informacji zawartych w KKT system drukuje, na żądanie, normy jednostkowe pracochłonności i materiałochłonności w różnych układach dla asortymentów wskazanych przez użytkownika. Dla każdego asortymentu można wydrukować również rozwinięcie /rozdetałowanie/ według poziomów montażu, a także zwinięcie czyli specyfikację zastosowania w asortymentach złożonych.

Ważną czynnością systemu jest założenie zbioru planu produkcji na podstawie dokumentu "Plan produkcji na rok..." oraz porównanie pracochłonności tego planu z możliwościami stanowisk pracy i ich obsad. Czynności te system wykonuje dla planu produkcji będącego portfelem zamówień oraz dla harmonogramu produkcji. System emituje bilanse zdolności produkcyjnej stanowisk i ich obsad dla okresów sprecyzowanych w miesiącach /od, do/ przez użytkownika. System umożliwia szybkie zbilansowanie zdolności produkcyjnej z wieloma wariantami planu. Dla każdej wersji planu emitowana jest, również na za-

danie użytkownika, informacja o potrzebach materiałowych dla jego realizacji, jak również wpływ zmian planu na potrzeby materiałowe. Dla ustalonej ostatecznej wersji planu system emituje dokumenty pobrania materiału /RW/ na materiały potrzebne do produkcji asortymentów dla poszczególnych zleceń produkcyjnych. Dla każdego zlecenia system emituje rozwinięcie konstrukcyjne według poziomów montażu, pokazując strukturę wyrobu od wierzchołka do podstawy, a także rozwinięcie sumaryczne zawierające ilości poszczególnych asortymentów potrzebnych do tego zlecenia.

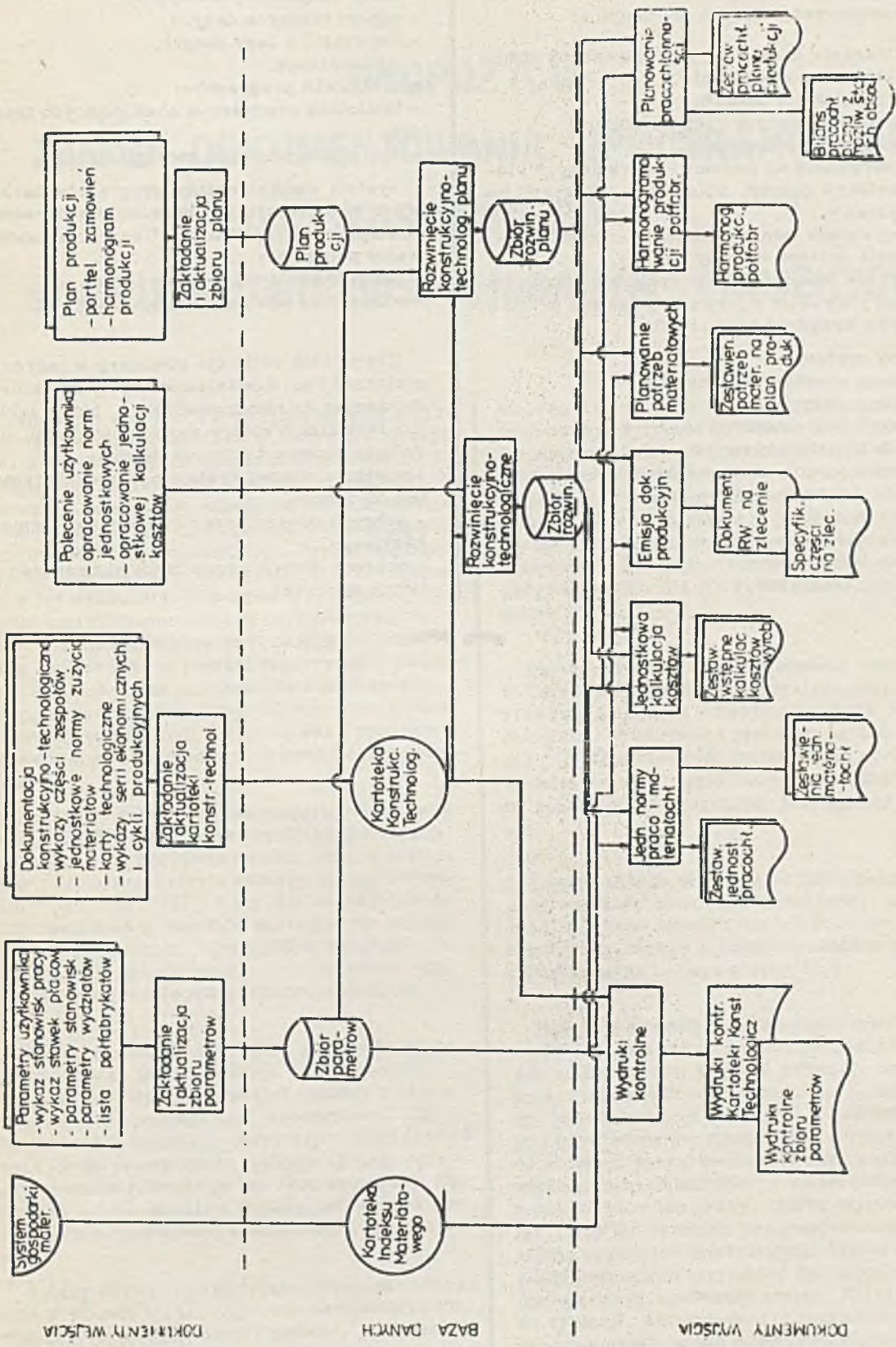
W systemie uwzględniono produkcję asortymentów na magazyn jako półfabrykatów. Na podstawie wyemitowanego przez system zestawienia asortymentów potrzebnych do realizacji przyjętego planu, użytkownik decyduje, które z nich będą produkowane jako półfabrykaty w danym roku kalendarzowym. Z listy tych asortymentów zakładany jest zbiór półfabrykatów. Korzystając z niego i biorąc pod uwagę wymogi planu, system układa harmonogram produkcji półfabrykatów z dokładnością do miesiąca, tak aby produkcja ich wyprzedzała co najmniej o miesiąc zapotrzebowanie na nie.

Dalszą czynnością systemu jest sporządzanie wstępnej jednostkowej kalkulacji kosztów dla żądanych asortymentów. Dodatkowe informacje potrzebne do tego celu użytkownik podaje na dokumencie "Żądanie kalkulacji jednostkowej". Przy wyżej wymienionych czynnościach system korzysta ze zbioru parametrów stanowisk, który zawiera takie informacje o stanowiskach jak: ilość maszyn, współczynnik wielowarsztatowości, % strat czasu stanowiska, % strat czasu obsady, średni % wyrobienia normy itp. oraz ze zbioru stawek placowych i ze zbioru parametrów wydziałów. Załączony schemat przedstawia budowę systemu.

System Wprowadzania i Wstępnego Przetwarzania Danych

System Wprowadzania i Wstępnego Przetwarzania Danych jest systemem narzędziowym dla EMC serii RIAD pracujących pod OS. Gromadzenie danych odbywa się poprzez sieć telekomunikacyjną w specjalnej bazie dokumentów. Sieć obsługiwana jest przez system SKOT i bazuje na końcówkach MERA 7910 sterowanych MERA 7904 lub MERA 7909. System w trybie zdalnym realizuje następujące funkcje:

- wprowadzanie danych, ich kontrolę formalną i merytoryczną w oparciu o zdefiniowane zbiory systemu; słownik danych, wzorce ekranów,
- przeglądanie bazy danych z możliwością automatycznego przejścia do trybu korekty danych,



Budowa systemu EPD. Normatywne Planowanie Środków Produkcji - SYNOPS

- usuwanie z bazy dokumentów zdezaktualizowanych,
 - przetwarzanie danych obejmujące:
 - wybieranie danych wg zdefiniowanych wzorców ekranów, wydawnictw,
 - porządkowanie danych,
 - akumulację danych i ich obróbkę,
 - prezentację wydawnictw.
 - generowanie na podstawie określonych kluczy selekcji danych, zbiorów zasilających inne systemy,
 - w systemie istnieją wielopoziomowe ograniczenia dostępu do bazy danych.
- Powyższe funkcje system realizuje w oparciu o zbiory struktur i parametry podane przez operatorów urządzeń końcowych.

Zbiory systemu:

- Zbiory struktur obejmujące:
 - słownik danych,
 - zbiór tablic odwzorowań,
 - zbiór wzorców ekranów,
 - zbiór wzorców dokumentów,
 - zbiór wzorców wydawnictw.
- baza danych:
 - zbiór indeksów dokumentów,
 - zbiór dokumentów,
- zbiory tranzytowe:

- log systemu,
- zbiór danych wejściowych,
- raport błędnych danych,
- "wyciągi" z bazy danych,
- wydawnictwa.
- biblioteka programów:
 - biblioteka programów obsługujących transakcje,
 - biblioteka procedur kontroli danych.

System stanowi zestaw programów narzędziowych umożliwiających generację konkretnego zastosowania /aplikacji/. Generacja zastosowania polega na:

- zdefiniowaniu zbiorów struktur,
- określeniu procedur kontroli danych.

Użytkownik definiuje struktury w języku formularzowym. Zdefiniowane opisy struktur podawane są działaniu procesora, który zakłada lub aktualizuje zbiory struktur systemu.

Zabezpieczenie i ochrona danych:

- wielopoziomowy system ograniczający dostęp do danych,
- ochrona danych przed awariami systemu operacyjnego,
- ochrona danych przed błędami urządzeń pamięci dyskowej.

PROPOZYCJA

ZASAD DOKUMENTOWANIA OPROGRAMOWANIA

STOSOWANEGO

W KOMPUTEROWO WSPOMAGANYM PROJEKTOWANIU

Niniejszy artykuł jest zwięzłą relacją z pracy wykonanej w roku 1983 w Instytucie Maszyn Matematycznych, na zlecenie OBR Podstaw Technologii i Konstrukcji Maszyn - TEKOMA. Założeniem pracy było zaproponowanie form dokumentowania oprogramowania wspomagającego projektowanie konstrukcji. Praca dotyczy formy i zawartości dokumentacji, a nie sposobów jej wykonania. Nie są to tym bardziej wymogi pod adresem samego oprogramowania. Tak więc proponowane tu rozwiązania mogą być realizowane w postaci tradycyjnej - drukowanej. Można też generowanie dokumentacji wspierać komputerowo. Można również analizować, jak ma być skonstruowany program z wbudowanymi elementami instrukcji użytkownika.

Jednym z elementów warunkujących rozwój metod KWP^{1/} jest sprawny obieg informacji o istniejącym oprogramowaniu, innym efektywne rozpowszechnianie samego oprogramowania [2], [8], [9] i szybkie zaznajomienie użytkowników w zakresie posługiwania się gotowymi produktami. Czynniki te wymagają m. in. ujednoczonych form dokumentowania produktów programowych przeznaczonych dla KWP.

Oprogramowanie dla KWP, jak każde oprogramowanie, już po oddaniu do eksploatacji, oprócz tego, że używane jest zgodnie z przeznaczeniem, podlega tzw. konserwacji, czy też raczej pielęgnacji. Przy czym musi istnieć możliwość prowadzenia pielęgnacji tego oprogramowania również przez "nie autorów". Będzie to jednak możliwe jedynie pod warunkiem istnienia odpowiedniej dokumentacji technicznej.

Należy dodać, że istniejące rozwiązania prawne w Polsce [4], [5] nie zaspokajają ww. potrzeb. Istniejące normy polskie, jak i ich

wzorce RWPG-owskie, szczegółowo scharakteryzowano w artykule Z. Hauswirta /Biuletyn Techniczno-Informacyjny MERA nr 1-2/1984 r. /. Generalnie i formalnie odnoszą się one do wszelkiego rodzaju oprogramowania komputerowego. W rzeczywistości jednak powstały z myślą o szeroko pojętym oprogramowaniu podstawowym. Nie odpowiadają więc specyfice oprogramowania dla KWP, są dla niego zbyt ogólne.

Faktem jest, że dokumentowanie tworzonych w Polsce oryginalnych i przystosowanych do rozpowszechniania w kraju produktów programowych, wykonywane jest z dużą dowolnością [4]. Ogólnikowość istniejących norm powoduje, że w zasadzie każdy, w miarę rozsądny sposób opisu programu, jest zgodny z normą.

Często zdarza się również takie celowe opracowywanie dokumentacji technicznej, aby unieemożliwić komukolwiek, oprócz autorów, poznanie budowy danego oprogramowania w stopniu pozwalającym na jego pielęgnację.

Należy podkreślić, że niedoskonałości te mają w dużym stopniu źródło w obecnej, bardzo niekorzystnej sytuacji prawnej, zmuszającej niejako autorów samego oprogramowania do "dbania o swoje interesy". Jeśli stworzono by odpowiednie podstawy prawne umożliwiające autorom oprogramowania czerpanie całkowite legalnych korzyści z wielokrotnego wdrożenia efektów ich pracy, byłiby zainteresowani tym, aby ich produkty programowe uzyskiwały status produktów powielanych. Status taki mogłoby produktom tym nadać np. uzyskanie dla danego oprogramowania atestu. Należy dążyć do sytuacji, aby autorom oprogramowania finansowo opłacało się dokumentowanie swoich produktów. Obecnie brak również dobrych, ogólnie przyjętych wzorców, czy też zasad opracowywania takiej dokumentacji.

1/ KWP - Komputerowo Wspomagane Projektowanie, odpowiednik angielskiego CAD.

Fazy życia programu a formy dokumentacji

Rozpatrując problemy oprogramowania dla KWP można wyróżnić następujące fazy życia tego oprogramowania [9]

- fazę tworzenia oprogramowania.
- fazę eksploatacji oprogramowania.

Niniejszy artykuł dotyczy fazy eksploatacji w tym sensie, że proponowane formy dokumentacji oprogramowania winny zapewnić prawidłowy przebieg tej fazy. Dotyczy również fazy tworzenia oprogramowania, gdyż opracowanie dokumentacji jest integralną częścią tej fazy [3], [5].

W fazie eksploatacji należy wyróżnić następujące aspekty zainteresowania się danym oprogramowaniem:

- Czy dane oprogramowanie może być przydatne dla danych problemów merytorycznych.
- Jak posługiwać się danym oprogramowaniem, aby sprawnie i optymalnie rozwiązać problem i w pełni wykorzystać możliwości przewidziane danym oprogramowaniem.
- Jak pielęgnować dane oprogramowanie.

Proponujemy więc stosować opisy:

1. Karta informacyjna.
2. Instrukcja lub podręcznik użytkownika.
3. Dokumentacja techniczna.

Należy dodać, że prace pielęgnacyjne przy oprogramowaniu polegają m. in. na uaktualnieniu odpowiednich fragmentów dokumentacji danego oprogramowania.

Rodzaje oprogramowania dla prac projektowych

Do oprogramowania dla KWP zalicza się takie produkty programowe, z którymi bezpośrednio pracuje projektant i które w treści swojej odnoszą się do problemów przez projektanta rozwiązywanych lub też wspomagają określone prace /fragmenty/ procesu projektowego.

W pracy [4] wprowadzono za [5] takie klasy oprogramowania, dla których będzie można zaproponować różniące się między sobą formy opisu programów. Zaproponowano też, aby forma Karty Informacyjnej była jednako dla wszystkich klas oprogramowania.

Instrukcję lub podręcznik użytkownika należy opracować odmiennie dla poniższych rodzajów oprogramowania:

- pojedyncze programy,
- pakiety programów problemowych i małe /proste/ systemy,
- biblioteki procedur /podprogramów/ problemowych,
- języki problemowe,
- systemy problemowe i podsystemy w systemach zintegrowanych.

Celem uściślenia pojęć poniżej przedstawiono definicję wymienionych rodzajów oprogramowania.

● Pojedynczy program - jest to w danym języku najmniejsza samodzielna jednostka programowa.

● Pakiet programów należy określić, jako pewną liczbę pojedynczych programów. Jeśli rozbuduje się pakiet, zapewniając silne związki między programami, wydziela się w osobne programy czynności wprowadzania i kontroli danych oraz generowania wyników; jeśli ponadto opracowuje się specjalne zbiory stałych danych projektowych, to można mówić o małym lub prostym systemie.

● Biblioteka procedur - jest to zbiór modułów programowych, mniejszych od programu; przy czym formalną definicję procedury przyjmuje się zgodną z definicją danego języka programowania. Są to np. dla ALGOLU i PASCALU "procedure", w FORTRANIE - "subroutine", "function" lub podobne przy innych językach kompilowanych. Natomiast przy językach interpretacyjnych, np. BASIC, stosuje się biblioteki programów. Biblioteki problemowe są dla inżynierów-projektantów dość trudne w zastosowaniu. Dlatego też często tworzy się, związany z daną biblioteką, język problemowy. Jest on zbliżony do języka danej dziedziny projektowania. Uwalnia użytkownika od znajomości języka algorytmicznego.

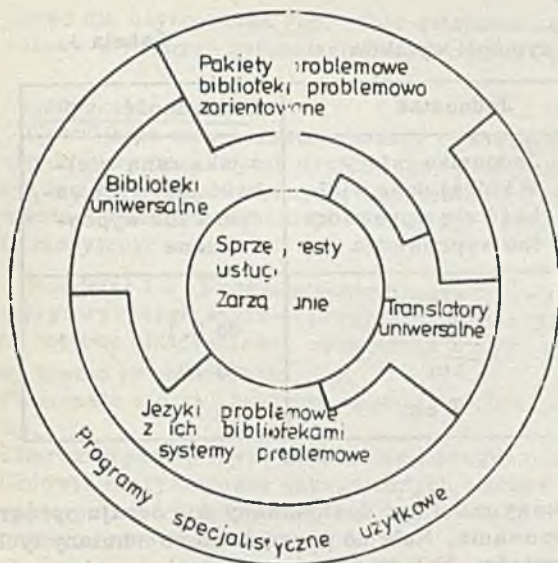
● Systemy problemowe i podsystemy problemowe w systemie zintegrowanym. Istnieje tu zawsze jakiś język problemowy. Język ten obejmuje procedury obliczeniowe, proste procedury wejścia i wyjścia, pozwalające na korzystanie ze zbiorów danych stałych danego systemu, na korzystanie i aktywne posługiwanie się zbiorami danych aktualnych, np. zbiorami, gdzie przechowuje się tzw. numeryczny model projektowania obiektu.

Formy dokumentacji użytkowej dla każdego ze scharakteryzowanych rodzajów oprogramowania, należy zróżnicować również w zależności od założonego sposobu eksploatacji danego oprogramowania. Forma dokumentacji technicznej zależy od ww. rodzajów oprogramowania i od elementów oprogramowania podstawowego danego komputera, na bazie których tworzone jest opisane oprogramowanie dla KWP. Jest to tzw. miejsce oprogramowania dla KWP w ogólnej strukturze oprogramowania /rys. 1/. Proponuje się więc dokumentację techniczną opracowywać odmiennie dla następujących rodzajów oprogramowania.

A. Produkty programowe pisane w języku algorytmicznym, z wykorzystaniem bibliotek uniwersalnych:

- pojedyncze programy,
- pakiety programów problemowych,
- biblioteki problemowe,
- małe /proste/ systemy.

B. Produkty programowe pisane jak w A, ale z wykorzystaniem bibliotek problemowych lub gotowych programów problemowych:



Rys. 1. Modelowa struktura oprogramowania dla KWP /5/

- pojedyncze programy,
- pakiety programów problemowych,
- języki problemowe,
- systemy problemowe.

C. Podsystemy problemowe pisane z wykorzystaniem narzędzi programistycznych systemu zintegrowanego.

Rodzaje dokumentacji

Karta informacyjna

Karta informacyjna danego oprogramowania winna zawierać informacje dla potencjalnego użytkownika zamierzającego nabyć dane oprogramowanie i dla decydenta, który ma podjąć decyzję o słuszności zakupu tego oprogramowania.

Proponujemy, aby w kasecie informacyjnej uwzględnić:

- nazwę formalną zwaną też handlową, czy symboliczną oraz krótką charakterystykę przeznaczenia i zakresu; użyte metody obliczeniowe: muszą być wymienione oprogramowania uzupełniające, informacje o poprzednich edycjach danego oprogramowania lub wzorcach czy analogach, inne referencje, zajętości pamięci, czasy typowych obliczeń;
- programistyczne i sprzętowe warunki, czyli:
 - pojemność pamięci operacyjnej,
 - ilość i rodzaj pamięci zewnętrznych,
 - urządzenia komunikacji z użytkownikiem pracujące on lub off line,
 - system operacyjny i procesory systemowe, tzw. utility, jak również ewentualnie język, w którym dane oprogramowanie jest napisane,
- autorzy, właściciele, rozpowszechniający i pielęgnowujący dane oprogramowanie, przy czym oprócz informacji o firmie celowe jest podanie nazwisk autorów danego produktu progra-

mowego i osób zobowiązanych do prac pielęgnacyjnych,

- data opracowania danego oprogramowania, ewentualnie dodatkowo, data ostatnio wprowadzonej poważnej modyfikacji lub rozszerzenia.

Jest sprawą dyskusyjną czy Karta Informacyjna powinna być wypełniona na jakimś z góry określonym formularzu, czy też powinna jedynie zawierać wszystkie wymienione treści. Jeżeli miałyby to być formularz proponujemy formę przedstawioną na rys. 2.

Opis użytkowy

Zasady ogólne

Ta forma opisu programu przeznaczona jest dla rzeczywistych użytkowników danego oprogramowania. Muszą tu być zawarte wszystkie niezbędne szczegóły merytorycznego zakresu obliczeń. Informacje te należy podawać w konwencji przyjętej w danej dziedzinie. Musi być podany zakres danych i charakterystyka generowanych przez oprogramowanie wyników.

W opisie użytkowym musi być bardzo dokładnie scharakteryzowany sposób posługiwania się oprogramowaniem - wyraźnie rozróżniający:

- sposób wywołania i uaktualniania oprogramowania jako całości lub poszczególnych jego fragmentów,
- sposób lub sposoby wprowadzania danych,
- sposób interpretacji otrzymanywnych wyników zasadniczych oraz wszelkiego rodzaju komunikatów i ostrzeżeń.

Karta informacyjna oprogramowania dla KWP			
Nazwa produktu programowego		Charakterystyka	
Uwarunkowania			
Konfiguracja sprzętu		Oprogramowanie	
Typ komputera		System operacyjny	
Pamięć operacyjna		Procesory systemowe	
Pamięć zewnętrzna		Język	
Urządzenie we/wy		Programy pomocnicze	
Autorzy: Firma:		Nazwiska:	
Właściciele: Firma			
Rozpowszechnia: Firma:		Nazwiska:	
Pielęgnowuje: Firma:		Nazwiska:	
Data opracowania:		Data modyfikacji:	
Atest	Numer	Data	Wydawjący

Rys. 2. Propozycja formularza Karty Informacyjnej oprogramowania dla KWP

Proponowana forma słownika symboli danych i symboli wyników

Tabela 1

Lp.	Symbol	Znaczenie	Jednostka	Dokładność
		Opis słowny, wzór powołanie się na wzór podany wcześ- niej w opisie itp.	Jednostka miary, w której daną wiel- kość się wprowadza lub wyprowadza	z jaką daną wiel- kość jest wprowa- dzana lub wypro- wadzana
4	P	siła obciążająca belkę	T	do 0,1
5	L	długość belki	cm	do 10
6	M	moment gnący	T cm	do 1

Muszą być również podane szczegółowe informacje o konfiguracji sprzętowej niezbędnej dla pracy oprogramowania, a także o możliwościach opcjonalnych lub wariantowych, jeśli takie istnieją. Autorzy przewidzieli np., że zasadniczo wynik będzie rysowany na plotterze, a jako rozwiązanie zastępcze, w przypadku braku plottera, wyprowadzane będą na drukarkę takie informacje, które pozwolą na ręczne wykreślenie odpowiedniego fragmentu.

Opis sposobów wprowadzania danych zależy od przyjętych rozwiązań programowych. Mogą to być opisy i wzory formularzy danych, opisy dialogów, itp. Zawsze konieczne jest podanie odpowiedniego "Słownika symboli" poszczególnych danych - proponowaną formę takiego słownika ilustruje tabela 1. Przy czym ten sam wzór powinien być zastosowany w odniesieniu do symboli pojawiających się w wynikach. Kolejność opisywanych symboli powinna być zgodna z kolejnością wprowadzania lub generowania wyników; natomiast przy "równoczesności" pojawiania się symboli, na rysunku lub ekranie monitora graficznego powinna być zachowana hierarchia ważności. Opis interpretacji wyników powinien zawierać odpowiednie przykłady wydruków, rysunków lub trwałych kopii dialogu.

Natomiast jako osobne zestawienie powinien być opracowany Wykaz Komunikatów /tabela 2/. Celem komunikatów może być ostrzeżenie użytkownika o nieprawidłowych lub nieprawidłowo wprowadzanych danych, zapytanie o wybór dalszej ścieżki obliczeniowej lub podobne. Komunikaty powinny być w zasadzie od razu generowane przez komputer w postaci zrozumiałej dla użytkownika - inżyniera. Zapewnienie wydruku odpowiedniego tekstu musi być rozwiązane w samym programie. Niemniej rozszerzenie znaczenia poszczególnych komunikatów jak i pełne ich zestawienie jest niezbędne.

W podręczniku użytkownika powinny się również znaleźć dobrze dobrane przykłady wszystkich czynności, jakie wykonuje użytkownik w trakcie posługiwania się programem. Opis użyt-

kowy musi być dostosowany do rodzaju oprogramowania. Kolejno przedstawiono odmiany tych opisów. Należy dodać, że jednakowe elementy opisu użytkowego wymienione są jednorazowo. Takimi jednakowymi elementami opisu użytkowego są:

1. Strona tytułowa

Na górze: nazwa instytucji firmującej dany program.

Pośrodku: nazwa programu i /jako podtytuł/ "Instrukcja" lub "Podręcznik użytkownika".

Na dole: miejscowość, rok.

2. Strona odwrotna strony tytułowej lub druga strona maszynopisu: adresy i telefony firmy i nazwiska osób - autorów, rozpowszechniających oraz prowadzących pielęgnację oprogramowania.

3. Spis treści opracowania.

4. Rozdział "Uwagi" występujący jako element końcowy. Autorzy opisu przedstawiają tu sprawy, które nie mogły być objęte żadnym z przewidzianych elementów opisu, a które, zdaniem autorów, są niezbędne lub przydatne dla prawidłowego użytkownika programu.

Ponadto proponuje się, aby o tym czy dany opis użytkowy nazwać "Instrukcją" czy "Podręcznikiem" ostatecznie decydowali autorzy opisu. Winni się oni kierować zasadą, iż Instrukcja jest opisem bardziej zwięzłym, syntetycznym, a podręcznik opisem bardziej rozbudowanym, nasyconym większą ilością przykładów, a zwłaszcza

Tabela 2

Proponowana forma wykazu komunikatów

Lp.	Tekst komunikatu	Interpretacje, sugestie
7	ZA DUŻĄ LICZBA SIŁ	dopuszczalna liczba sił 8, rozdzielić na części i powtórzyć obliczenia

porad dla użytkownika /np. jakie działania są celowe w różnych sytuacjach projektowych itp. /.

Pojedynczy program

Rozwijając sformułowane zasady odnoszące się do zawartości opisu użytkowego, proponujemy, aby po trzech wstępnych, wymienionych elementach następowały kolejno rozdziały i elementy merytoryczne.

Rozdział 1 - "Przeznaczenie programu" - opis merytorycznego zakresu programu, wykorzystane metody obliczeniowe, zasadnicze wzory, opisy metod przetwarzania, itp.

Powołanie się na literaturę, prace źródłowe, itp.

Charakterystyka trybu eksploatacji programu. Omówić merytoryczny zakres danych i scharakteryzować w sposób ogólny, generowane przez program wyniki.

Rozdział 2 - "Warunki działania programu" - szczegółowy opis konfiguracji sprzętowej niezbędnej dla prawidłowej pracy programu: typ komputera, potrzebne dla pracy programu rozmiary pamięci operacyjnej, rodzaj i liczba jednostek pamięci zewnętrznych, urządzenia wejścia i wyjścia, z zaznaczeniem ew. urządzeń wymaganych jedynie opcjonalnie, szczegółowy opis wymaganego oprogramowania uniwersalnego, system operacyjny, "utility", tłumaczący, biblioteki uniwersalne, itp. Należy też podać charakterystyczne czasy działania programu w warunkach typowych oraz ekstremalnych.

Rozdział 3 - "Uaktywnianie programu" - czynności, które muszą być wykonane, aby program był gotowy do działania. Należy wyraźnie powiedzieć kto i które z tych czynności winien wykonywać. Opisy te będą różne w zależności od sposobu eksploatacji programu. Przy trybie wsadowym wszystkie czynności wykonuje operator. Przy trybie konwersacyjnym i interaktywnym większość czynności uaktywnienia wykonuje bezpośredni użytkownik pracujący przy końcówce - terminalu. Jeśli program jest uaktywniany /"wołany"/ przez użytkownika pracującego z monitorem, należy podać kolejność włączania klawiszy funkcyjnych oraz dokładny format tekstów, które muszą być wypisane na monitorze. Celowe jest pokazanie rysunku klawiatury z zaznaczeniem klawiszy funkcyjnych i kolejności ich użycia.

Rozdział 4 - "Wprowadzanie danych" - dokładny opis danych, kolejność i sposób ich wprowadzania. Opis danych powinien być przedstawiony w formie tzw. słownika symboli danych /tabela 1/. Jeśli program eksploatowany jest w trybie wsadowym, musi być też pokazany wzór formularza danych /rys. 3/; w tym przypadku należy również podać jak na podstawie wypełnionych formularzy mają być przygotowywane maszynowe nośniki danych i jak mają one być wprowadzane do komputera. Konieczne jest omówienie reakcji programu na wprowadzane

dane i na ewentualne błędy w danych. Takimi możliwymi reakcjami są: przedruk danych, komentarze o akceptacji danych, komunikaty o błędach, wstrzymanie pracy programu, itp. Przy opisie wprowadzania danych dla programów eksploatowanych wsadowo, muszą być wyraźnie rozgraniczone czynności, które wykonuje użytkownik i czynności, które wykonywane są w ośrodku obliczeniowym.

Jeśli program przewiduje konwersacyjne wprowadzanie danych należy po "Słowniku symboli danych" przedstawić pełny opis dialogu. Muszą być więc podane kolejno wszystkie komunikaty, które generuje komputer i wszystkie możliwe warianty tekstów wprowadzanych przez użytkownika. Należy omówić konsekwencje każdego wariantu wprowadzanej przez użytkownika informacji.

Jeśli program przewiduje wprowadzanie danych graficznych muszą być omówione wszystkie związane z tym czynności. I tak, jeśli jest to wprowadzanie danych z digitizera - należy opisać kolejność działań, dopuszczalne formaty rysunków, dopuszczalną dokładność, czynności konieczne dla prawidłowego wygenerowania nośnika danych itp. Wyraźnie powinny zostać wydzielone czynności wprowadzania tych danych z nośnika do komputera, czyli do programu. Jeśli jest to wprowadzanie rysunku z monitora graficznego - muszą być opisane wszystkie możliwe, czyli wszystkie akceptowane przez program, formy rysunków i niezbędne rozkazy.

Rozdział 5 - "Generacja wyników" - omówienie wszystkich form generowanych przez program, ostatecznych wyników. A więc możliwe teksty wydruków - opisów, formy tablic, rysunki i ich opisy. Należy podać interpretację każdej z form oraz odpowiednie przykłady. W rozdziale tym konieczne jest również umieszczenie "Słownika symboli wyników" - opracowanego zgodnie z przedstawionymi /tabela 1/ zasadami. Przy eksploatacji programu w trybie wsadowym należy podać "Wykaz komunikatów" - są to wszelkie teksty wyprowadzane podczas pracy programu, a nie wchodzące w zakres wyników ostatecznych /końcowych/. Przy eksploatacji programu w trybie konwersacyjnym w zasadzie komunikaty tego typu powinny być ujęte w opisie dialogowego wprowadzania danych. Jeśli jednak program jest tak zorganizowany, że poza fragmentem dialogu, odnoszącym się ściśle do wprowadzania danych, są fragmenty dialogu odnoszące się do pracy programu i realizujące interwencję użytkownika w przebiegu programu, wówczas opisy tych dialogów można przedstawić oddzielnie.

Przykłady:

Testy - są to specjalnie dobrane zestawy danych, które pozwalają użytkownikowi sprawdzić poprawność pracy programu. Muszą być tu zamieszczone komplety komunikatów i wyników ostatecznych dla tych danych.

..... OBIEKT
 NR PODPORY
 NR FORMULARZA

..... JEDNOSTKA ZLECAJĄCA OBLICZENIA
 NR OBLICZEŃ
 NR ARCH.

FORMULARZ DANYCH

DO PROGRAMU "PODPORA ZE SŁUPAMI FUNDAMENTOWYMI ZWIĘNZIONYMI PŁYTA NIEODKSIĘTLIWA"

		NUMERY SŁUPÓW W PODPORZE																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
e3	ns																		
	x																		
	om																		
0	444 333	/																	
	e																		
	d																		
	ds																		
	lx																		
	EJ																		
	EF																		
	hg																		
	lo																		
	C																		
	m																		
1	444 333																		
e1	444 333	e	d	ds	lx	EJ	EF	hg	lo	C	m	12000	200	144 333					

PRZY WIĘKSZEJ LICZBIE SŁUPÓW FORMULARZ PRZEDLUŻYC

		WARIANTY OBLICZEN																							
		P	Mx	Hx	dI	n6	n5	n4	n1	n2	n														
					444 333	P	/	/	/	/	/	/	/	/	/	/	/								
					444 333	P	/	/	/	/	/	/	/	/	/	/	/								
					444 333	P	/	/	/	/	/	/	/	/	/	/	/								
					444 333	P	/	/	/	/	/	/	/	/	/	/	/								
					444 333	P	/	/	/	/	/	/	/	/	/	/	/								

UWAGI:

 OPRACOWAŁ DATA
 SPRAWDZIŁ DATA

Rys. 3a.

Przy eksploatacji wsadowej należy załączyć przykłady wypełniania formularzy danych, jak też przykłady rysunków możliwych do digitalizacji, pokazujących możliwe warianty.

Przy eksploatacji w trybie konwersacyjnym należy podać przykłady wszystkich wariantów poszczególnych fragmentów dialogów - są to tzw. kopie sesji. Szczególnie ważne są przykłady wprowadzania danych z monitora graficznego, powinny to być zdjęcia monitora.

Pakiet programów i mały system

Po pierwszych jednakowych elementach opisu użytkowego, następne elementy są specyficzne dla obu typów oprogramowania.

Rozdział 1 - powinien nosić tytuł "Przeznaczenie i struktura pakietu" /systemu/. Należy tu podać merytoryczny zakres całego oprogramowania, przyjęte metody obliczeniowe, stosowane wzory. Następnie trzeba wymienić, jakie programy wchodzi w skład pakietu czy systemu, jakie są ich zasadnicze, merytoryczne zakresy oraz czy istnieją powiązania lub zależności między poszczególnymi programami i na czym one polegają. Należy scharakteryzować, jakie obliczenia można wykonać przy pomocy poszczególnych programów i ich odpowiednich zestawów.

Konieczne jest podanie opisu zakresu danych, które musi dostarczyć użytkownik, posługując się odpowiednimi zestawami programu i jakie otrzyma ostateczne wyniki: oddzielnie należy opisać, jakie dane potrzebne są do pracy poszczególnych programów i jakie wyniki poszczególne programy generują. Należy też podać rodzaje zbiorów danych, a mianowicie zbiory robocze, stałe zbiory danych aktualnych i ewentualnie stałe zbiory danych statycznych.

Następne rozdziały opisu użytkowego pakietu programów problemowych lub małego systemu zatytułowane są podobnie jak w opisie pojedynczego programu i zawierające analogiczną treść. Są to:

Rozdział 2 - "Warunki działania pakietu" /lub "systemu"/, w którym zawarto niezbędne dla użytkownika informacje o konfiguracji sprzętu i oprogramowania podstawowego.

Rozdział 3 - "Uaktywnienie pakietu" /"systemu"/ - sposoby uaktywniania poszczególnych pojedynczych programów i poszczególnych grup programów - dla tych ostatnich szczególnie ważne jest zwrócenie uwagi na ewentualną wymaganą ze względów merytorycznych, kolejność uaktywniania programów w danej grupie. Jeśli opisywanym oprogramowaniem jest mały system, a więc występują "stałe zbiory danych", należy wydzielić opis uaktywniania programów zakładania tych zbiorów oraz opis uaktywniania programów, modyfikowania zbiorów "stałych danych projektowych". Czynności odnoszące się do "stałych zbiorów danych" wykonywane są w innym czasie niż uaktywnianie oprogra-

mowania wykonującego obliczenia projektowe i posługującego się informacją mieszczącą się w tych zbiorach.

Rozdział 4 - "Wprowadzanie danych". Jeśli oprogramowanie jest pakietem programów problemowych, to opis wprowadzania danych jest taki jak dla pojedynczego programu. Musi być jedynie zwielokrotnione, czyli powtórzone w odniesieniu do poszczególnych programów. Jeśli opisanie oprogramowania jest małym systemem, trzeba osobno opisać sposoby wykonania wszystkich czynności niezbędnych dla wprowadzania odpowiednich danych do poszczególnych zbiorów. Należy odrębnie omówić wprowadzanie danych do zbiorów przy ich zakładaniu oraz wprowadzaniu modyfikacji i poprawek, itp. Niezbędne jest również zilustrowanie ww. czynności odpowiednimi schematami.

Pozostałe elementy opisu użytkowego są analogiczne jak przy opisie pojedynczego programu. Są to: Rozdział 5 - "Generowanie wyników" wraz z "Wykazem komunikatów" lub opisem dialogów" oraz "Przykłady".

Biblioteka procedur problemowych

Opis użytkowy tego typu oprogramowania powinien zawierać, oprócz wymienionych w "Zasadach ogólnych" /str29/ stałych, następne wyraźnie odmiennie elementy.

Rozdział 1 - "Ogólna charakterystyka biblioteki" - opis zasadniczego zakresu problemów, do rozwiązania których można wykorzystywać procedury opisywanej biblioteki. Należy podać język, w jakim powinny być pisane programy wywołujące poszczególne procedury, niezbędną konfigurację sprzętową, wymagany system operacyjny, niezbędne oprogramowanie typu "utility", biblioteki uniwersalne itp.

Rozdział 2 - "Opis procedur" - wymienić kolejno wszystkie procedury, podając o każdej następujące informacje:

- nazwa procedury,
- funkcja, jaką dana procedura realizuje,
- algorytm, wzory, ewentualnie powołanie się na literaturę,
- sposób wołania, wykaz parametrów formalnych, opis ich znaczenia,
- dane: wymienić, które parametry formalne pełnią rolę danych, typ każdej, zakres, ewentualnie jakie dane do procedury wprowadzane są inaczej,
- wyniki: opis według analogicznych zasad jak dane,
- procedury współpracujące: jeśli dla pracy opisywanej procedury konieczne jest w programie głównym użycie innych procedur należy podać wszystkie informacje potrzebne dla piszącego program użytkowy,
- rozmiar procedury - ilość instrukcji, słów lub podobne informacje ważne dla piszącego program.

"Przykłady": pokazać sposoby budowania programu z użyciem procedur opisywanej biblioteki.

Język problemowy

Opis użytkowy oprócz elementów stałych winien zawierać elementy specyficzne.

Rozdział 1 - "Przeznaczenie języka" - opis merytoryczny problemów, jakie można rozwiązać przy pomocy tego języka, czyli dziedzina zastosowań, klasy problemów.

Rozdział 2 - "Ogólna charakterystyka języka" - opis struktury języka czyli podstawowych zasad, według których język korzysta z odpowiedniej biblioteki problemowej i ogólną postać rozkazów języka. Jeżeli w języku występują zasadnicze różne struktury rozkazów, należy scharakteryzować każdą z tych struktur. Należy wyraźnie powiedzieć, w jakim trybie przewidziano korzystanie z języka - wsadowym czy konwersacyjnym oraz wstępnie scharakteryzować zasady pisania programów w opisywanym języku. W rozdziale tym niezbędne jest również omówienie sposobów postępowania operatora, przy klasycznym trybie wsadowym lub osoby pełniącej jego funkcje, przy innym trybie - konieczne dla uaktywnienia oprogramowania "Język problemowy". Ten fragment opisu powinien być, co do swojej formy, zgodny z rozdziałem "Uaktywniania programu" omówionym na str. 31.

Rozdział 3 - "Warunki pracy języka" - opis konfiguracji - sprzętu i oprogramowania niezbędnych dla tworzenia i eksploataowania programów napisanych w danym języku problemowym.

Rozdział 4 - "Szczegółowa charakterystyka rozkazów języka". Należy tu wymienić kolejno wszystkie rozkazy języka, podając następujące informacje:

- ogólna postać danego rozkazu,
- zasadnicze przeznaczenie /funkcja/,
- zakres działania każdego z ewentualnych wariantów /rozgałęzień/,
- szczegółowa postać danego wariantu wraz z listą zmiennych, ich znaczeniem, formatem i zakresem,
- komunikaty, jakie mogą być wygenerowane w wyniku działania danego rozkazu.

Rozdział 5 - "Zasady korzystania z języka" - sposoby tworzenia w opisywanym języku problemowym, programów użytkowych, obowiązująca /obligatoryjna/ sekwencja rozkazów i możliwe wariantowania. Jeżeli posługiwanie się językiem realizowane jest w trybie wsadowym, należy omówić, które czynności wykonuje użytkownik, a które obsługa ośrodka obliczeniowego. Jeżeli natomiast tworzenie programu użytkowego ma być realizowane w trybie konwersacyjnym, należy zwrócić uwagę użytkownika na te czynności, które związane są z uruchamianiem stworzonego w języku problemowym programu użytkowego, a więc ładowanie do komputera, translowanie, poprawianie błędów.

Rozdział 6 - "Wyniki programów użytkowych" - winien omawiać formy generowania, przez programy użytkowe napisane w języku problemowym, ostatecznych wyników przydatnych w procesie projektowym. W tym opisie należy się odwoływać do szczegółowych charakterystyk odpowiednich rozkazów języka zamieszczonych w rozdziale 5 opisu danego języka.

Rozdział 7 - "Opis metod" - należy zamieścić opis wszystkich metod merytorycznych danej dziedziny inżynierii, które zostały przyjęte w bibliotece problemowej, będącej podstawą dla opisywanego języka problemowego. Podać algorytmy, wzory, itp. zasady obliczeniowe, łączyć je z odpowiednimi rozkazami języka, nie podając natomiast nazw procedur ani ich budowy.

"Przykłady" - wskazane jest zamieszczenie przykładowych tekstów programów napisanych w języku problemowym wraz z przykładami wstawienia konkretnych danych i wygenerowanymi przez te programy wynikami.

System problemowy i podsystem w systemie zintegrowanym

Opis użytkowy zawiera poza elementami stałymi opis języka problemowego oraz opis zasad tworzenia i obsługi zbiorów.

Rozdział 1 - "Przeznaczenie systemu" - ma scharakteryzować obszar zagadnień czyli dziedzinę inżynierii, której problemy można rozwiązywać przy pomocy opisywanego oprogramowania.

Rozdział 2 - "Warunki działania systemu" - należy wymienić niezbędną konfigurację sprzętu i oprogramowania, przy czym o ile opisywany system problemowy był tworzony jako podsystem pewnego systemu zintegrowanego, należy opisać środki programowe tego systemu zintegrowanego konieczne dla pracy opisywanego systemu problemowego.

Rozdział 3 - "Struktura systemu" - winien scharakteryzować strukturę języka problemowego według zasad przedstawionych w poprzednim rozdziale artykułu "Język problemowy". Ponadto należy scharakteryzować zbiory danych stałych i aktualnych oraz oprogramowanie służące do tworzenia tych zbiorów i oprogramowanie dla prowadzenia danych stałych i ich modyfikacji.

Rozdział 4 - "Obsługa stałych zbiorów danych" - winien opisać sposoby uaktywniania poszczególnych programów obsługi zbiorów według zasad podanych w rozdziale "Uaktywnienie programu". Następnie scharakteryzować rodzaje danych, które mają być wprowadzane do każdego z tych zbiorów, sposób ich przygotowania przez użytkownika, zasady tworzenia nośników maszynowych i wszystkie czynności operatorskie niezbędne dla wprowadzania stałych danych projektowych do odpowiednich zbiorów. Wyraźnie należy zastrzec, że programami obsługi stałych zbiorów danych,

posługują się, bądź pracownicy ośrodka, bądź odpowiednio wyspecjalizowani użytkownicy, czyli tzw. "obsługa systemu". Natomiast korzystanie zarówno bierne ze zbiorów stałych danych projektowych, jak i czynne i bierne ze zbiorów danych aktualnych realizowane jest jedynie w ramach języka problemowego. W rozdziale tym konieczne jest również podanie wszystkich wykazów komunikatów generowanych przez programy obsługi stałych zbiorów danych.

Następne rozdziały opisu użytkowego przeznaczone są dla użytkowników języka problemowego systemu i powinny być napisane zgodnie z zasadami przedstawionymi w rozdziałach: 5 - "Szczegółowa charakterystyka rozkazów języka", 6 - "Zasady korzystania z języka", 7 - "Wyniki programów użytkowych", 8 - "Opis metod". Natomiast w ramach "Przykładów" należy podać zarówno przykłady tekstów programów napisanych w języku problemowym wraz z przykładami wygenerowanych przez te programy wyników, jak i wyraźnie wydzielone przykłady posługiwania się programami obsługi zbiorów danych stałych.

Opis techniczny

Zasady ogólne

Dokumentacja techniczna musi zawierać kompletne i szczegółowe wiadomości o budowie i strukturze danego oprogramowania, zarówno jako całości, jak i o wszystkich jego częściach. Opis ten powinien być ilustrowany odpowiednimi schematami charakteryzującymi: zasadniczą strukturę oprogramowania, z podziałem na główne moduły, a następnie prezentując coraz szczegółowsze opisy modułów i innych struktur oprogramowania, np. zbiorów danych. Powinno się tu również omówić dlaczego przyjęto takie,

Tabela 4

Proponowana forma alfabetycznego indeksu symboli całego oprogramowania

Lp.	Symbol	Miejsce występowania
1	AS	Wymienić wszystkie modu-
2	IZ	ły, w których ten symbol
3	LICZ	jest użyty, z ew. poda-
4	R	niem wspólnego bloku.

a nie inne rozwiązanie organizacyjne i programowe, a nawet, czy i jakie inne rozwiązania rozważano i dlaczego z nich zrezygnowano.

Opisując fragmenty obliczeniowe wskazane jest dokładne sprecyzowanie jaką przyjęto metodę, i jak ją zalgorytmizowano na użytek tego oprogramowania, z ewentualnym powołaniem się na opracowania źródłowe /publikacje naukowe, przepisy techniczne itp. /. Opis poszczególnych, używanych w programie symboli musi być opracowany bardzo jasno. Nie wolno lekceważyć opisu symboli lokalnych i roboczych. Dla symboli używanych w różnych częściach oprogramowania należy fakt ten wyraźnie zaznaczyć. Warto też zwrócić uwagę, jakie problemy związane ze zmiennymi mogą wystąpić przy pielęgnacji oprogramowania. Opisy symboli winno się zestawiać w słowniki symboli /tabela 3/, w których symbole należy ustawić wg pewnej hierarchii ważności. Przy obszernym oprogramowaniu, słowniki te trzeba opracowywać oddzielnie dla poszczególnych modułów oprogramowania. W takiej sytuacji należy opracować alfabetyczny indeks symboli całego oprogramowania /tabela 4/.

Tabela 3

Proponowana forma słownika symboli w dokumentacji technicznej

Nazwa modułu, typ, znaczenie, zakres, inne informacje o module					
Lp.	Symbol	Typ	Rodzaj	Charakter	Znaczenie, ew. sugestie pielęgnacyjne
1	IZ	int.	zm. pr.	dana	fizyczne znaczenie danego symbolu, ew. powołanie się na wzór merytoryczny lub podanie wzoru roboczego. Gdy ze wspól. bloku, to którym zmiennym z innych modułów odpowiada ta zmienna itp. Gdy zm. rob. ma parę znaczeń w danym module, to podać wszystkie.
2	R	real	tabl.	wynik	
3	AS	.	nazwa	par. form.	Gdy ze wspól. bloku, to którym zmiennym z innych modułów odpowiada ta zmienna itp. Gdy zm. rob. ma parę znaczeń w danym module, to podać wszystkie.
4	LICZ	.	wsp. bloku	zm. lok.	
		.	nazwa proc.	zm. rob. jeżeli ze wsp. bl. to jego nazwa	
		.	.	.	

W dokumentacji technicznej należy dużo uwagi poświęcić opisowi powiązań dokumentowanego oprogramowania z innym oprogramowaniem, np. z systemem operacyjnym i używanymi programami typu "utility", z językiem, w jakim program został napisany, z bibliotekami uniwersalnymi i problemowymi. Sprawy te mają istotny wpływ na prace pielęgnacyjne, jak i na przenoszalność danego oprogramowania. Dokumentacja techniczna musi zawierać pełne wydruki źródłowego tekstu oprogramowania, zaopatrzone w odnośniki do słownych i graficznych opisów.

Osobnym rozdziałem dokumentacji technicznej są opisy sposobów testowania prawidłowej pracy oprogramowania, zawierające wydruki wyników testowania. W dokumentacji technicznej pokazane jest zasygnalizowanie rozszerzenia dokumentacji technicznej lub modyfikacji wraz z ostrzeżeniem, iż pewne zmiany mogą wywołać określone konsekwencje. Należy zaznaczyć, które rozwiązania przyjęto z uwagi na warunki sprzętowe lub programowe, i jak, przy zmianach tych warunków, należałoby zmienić dane oprogramowanie. W odniesieniu do dokumentacji technicznej jeszcze silniej, niż w stosunku do opisu użytkownika, występuje zależność formy dokumentacji od rodzaju oprogramowania. Warianty formy dokumentacji technicznej muszą być jednak wszystkie zgodne z ogólnymi zasadami. Warianty te będą kolejno przedstawione, w przypadku gdy pewne elementy opisu technicznego będą takie same, nie będzie się ich szczegółowo wielokrotnie powtarzać. Takimi identycznymi elementami są:

1. Strona tytułowa - która winna być identyczna ze stroną tytułową opisu użytkowego, jedynie jako podtytuł, nazwy programu, należy umieścić: "Dokumentacja techniczna".
2. Strona odwrotna strony tytułowej - identyczna, jak dla opisu użytkowego.
3. Spis treści opracowania.
4. Rozdział - "Zalecenia i sugestie" występujący jako końcowy element zawiera informacje dotyczące zabiegów pielęgnacyjnych.

Należy również przyjąć zasadę niepowtarzania w dokumentacji technicznej informacji wyczerpująco podanej w opisie użytkowym, omawiając zakres funkcjonalny i przeznaczenie danego oprogramowania ujęte w Rozdziale 1 Dokumentacji Technicznej, należy podać tylko zasadnicze informacje, odsyłając równocześnie do opisu użytkowego.

Pojedynczy program

Wersja a - program pisany jako oprogramowanie bezpośrednio.

Rozdział 2 - "Struktura programu" - zaprezentowana w postaci schematu organizacyjnego, na którym rozróżnia się zakresy kolejnych czynności realizowanych przez program. O ile opracowano procedury /podprogramy/ swoiste dla opisywanego programu, to na sche-

macie organizacyjnym należy pokazać je jako zwarte bloki, jednak bez pokazywania ich wewnętrznej budowy. Ta wewnętrzna budowa, zarówno poszczególnych procedur, jak i ewentualnych pewnych bloków które formalnie nie są procedurami, ale pełnią wydzielone funkcje w ramach opisywanego programu należy pokazać na osobnych schematach. W rozdziale tym należy wyjaśnić, dlaczego przyjęto właśnie taką strukturę programu oraz podać, czy rozważano inne możliwości, na czym one polegały i dlaczego z nich zrezygnowano. Aby ocenić słuszność przyjętych rozwiązań tego typu informacje będą przydatne zarówno przy "odbiorze", jak i przy "atestacji" oprogramowania. Mogą być też niezbędne przy prowadzeniu zabiegów pielęgnacyjnych. Przy opisach fragmentów obliczeniowych /lub innych realizujących czynności merytoryczne/ można odsyłać do opisu użytkowego, ale w opisie technicznym należy podawać, dlaczego opisane tam metody zalgorytmizowano właśnie w taki, a nie w inny sposób.

Rozdział 3 - "Opisy symboli" - wszystkie symbole występujące w programie należy zestawić w odpowiedni słownik zgodnie z przedstawionymi zasadami /tabela 3, 4/.

Rozdział 4 - "Uwarunkowania" - omówić te wszystkie elementy sprzętu i oprogramowania, które były wykorzystywane przy tworzeniu opisywanego programu i które mogą być przydatne przy jego pielęgnacji. Środki te mogą być odmienne od tych, które są niezbędne dla użytkownika programu.

"Tekst programu w języku źródłowym". - powinien być to oryginalny wydruk przekazywanej do eksploatacji wersji programu.

"Wyniki przebiegów testowania" - powinny być to pełne wydruki z przebiegów pracy programu dla odpowiednich zestawów danych testowych, czyli - wydruk danych, wydruki komunikatów, wyniki.

Wersja b - program pisany jako oprogramowanie pośrednie otwarte, tzn. z użyciem zasobów odpowiedniej biblioteki problemowej.

W rozdziale 2 - "Struktura programu" należy wymieniły użyte podprogramy z danej biblioteki problemowej. Natomiast w dalszych częściach tego rozdziału, gdzie podawane są dokładne struktury i funkcje wszystkich części opisywanego programu należy jedynie odwołać się do opisu używanej biblioteki.

Rozdział 3 - "Opis symboli" - wyjaśnić dla symboli będących parametrami aktualnymi procedur pochodzących z biblioteki problemowej, jakim parametrom formalnym tych procedur odpowiadają.

Rozdział 4 - "Uwarunkowania" - omówić, z jakiej biblioteki lub bibliotek problemowych korzystano przy pisaniu programu oraz powo-

łać się na odpowiedni opis użytkowy lub techniczny tej biblioteki.

Pozostałe elementy dokumentacji technicznej pojedynczego programu w wersji b są identyczne, jak w wersji a.

Pakiet programów problemowych

Wersja a - dla pakietu opis struktury powinien dzielić się na następujące części:

Rozdział 2 - "Ogólna struktura pakietu" - rozróżniana z dokładnością do pojedynczego programu. Wskazać na merytoryczne powiązania między programami np. wyniki pracy jednego programu są danymi dla drugiego, itp. oraz zaznaczyć niezależność lub wymienność poszczególnych programów. To ostatnie odnosi się np. do sytuacji gdy w skład pakietu wchodzi programy rozwiązujące ten sam problem, ale różnymi metodami. Strukturę pakietu należy zilustrować schematem. Wskazane jest również omówienie przyjętych zasad komunikacji między programem, np. przez zbiór dyskowy roboczy lub inne.

Rozdział 3 - "Szczegółowe struktury programów" - składa się z tyłu podrozdziałów, ile wchodzi programów w skład pakietu a każdy powinien zawierać treść zgodną z zasadami omówionymi w rozdziale "Struktura programu".

Rozdział 4 - "Opisy symboli" - składa się z tyłu podrozdziałów, ile programów liczy pakiet a każdy z tych podrozdziałów, powinien być napisany zgodnie z odpowiednimi zasadami zaprezentowanymi dla pojedynczego programu.

Rozdział 5 - "Uwarunkowania" - co do formy powinien być identyczny z odpowiednim rozdziałem omawianym dla pojedynczego programu.

"Teksty programów w języku źródłowym" - powinny zawierać wydruki każdego z programów, wchodzących w skład pakietu.

"Wyniki przebiegów testowania" - powinny pokazywać zarówno wyniki testowania każdego oddzielnie programu /zgodnie z zaprezentowanymi dla pojedynczego programu zasadami/, jak i wyniki testowania wszystkich możliwych grup programów, celem udokumentowania prawidłowości przyjętych sposobów przekazywania informacji między programami.

Wersja b - w opisie technicznym pakietu, fakt korzystania z biblioteki procedur problemowych ma swoje odzwierciedlenie w rozdziale 3 "Szczegółowe struktury programów" - przy opisie poszczególnych programów należy kierować się zasadami dla pojedynczego programu.

Wersja b - podobny zabieg należy zastosować w rozdziale 4 - "Opisy symboli", gdzie dla każdego programu, w którym skorzystano z procedur biblioteki problemowej, należy omówić parametry aktualne tych procedur ze

wskazaniem odpowiednich parametrów formalnych. Również w rozdziale 5 - "Uwarunkowanie" należy podać z jakiej biblioteki problemowej skorzystano oraz powołać się na odpowiedni opis użytkowy tej biblioteki.

Pozostałe elementy dokumentacji technicznej pakietu programów problemowych dla wersji b są identyczne, jak dla wersji a.

Biblioteka procedur problemowych

Oprogramowanie tego typu jest zawsze oprogramowaniem bezpośrednim. Ponadto z definicji biblioteki wynika, że nie można mówić o strukturze całego oprogramowania. Dokumentacja techniczna całej biblioteki poza elementami jednakowymi winna więc składać się z "Opisów technicznych poszczególnych procedur". Opis techniczny poszczególnych procedur powinien, oprócz odwołania się do opisu użytkowego tej procedury, gdzie podana jest funkcja i zakres merytoryczny/, zawierać następujące elementy:

- schemat blokowy.
- opis wszystkich symboli.
- tekst procedury.
- wyniki testowania.
- zalecenia i sugestie dla zabiegów pielęgnacyjnych.

Opis techniczny biblioteki jako całości, powinien również zawierać rozdział 3 danego opisu "Uwarunkowania", opracowany zgodnie z zasadami podanymi dla pojedynczego programu.

Mały system

Zgodnie z charakterystyką małego systemu jako pakietu programów problemowych oraz stałych zbiorów danych projektowych /stałych i aktualnych/, dokumentacja techniczna tego oprogramowania winna zawierać wszystkie elementy opisane dla pakietu programów odpowiednio dla wersji a/ - jeżeli jest to oprogramowanie bezpośrednie lub dla wersji b/ - jeżeli jest pisane jako pośrednie, jednak zawsze rozbudowane o odpowiednią informację o zbiorach. Tak więc:

Rozdział 2 - "Ogólna struktura systemu" - opracować wg zasad podanych dla pakietu z koniecznym uwzględnieniem opisu zbiorów stałych. Należy podać nazwy zbiorów i ich przeznaczenie, zaś strukturę systemu zilustrować schematem, na którym uwidocznione będą wszystkie elementy systemu: oprogramowanie problemowe, stałe zbiory danych, oprogramowanie dla obsługi tych zbiorów.

Rozdział 3 - "Szczegółowa struktura programów" - opracować wg zasad podanych dla pakietu, dla wszystkich programów wchodzących w skład systemu, a więc dla programów problemowych i dla programów obsługi zbiorów stałych, z zachowaniem wymagań wersji a/ lub b/ w zależności od sytuacji.

Rozdział 4 - "Opisy symboli" - opracować według zasad dla pakietu, uwzględniając zarówno programy problemowe, jak i programy obsługi zbiorów.

Rozdział 5 - "Szczegółowy opis zbiorów", dokładnie omówić poszczególne zbiory i ich strukturę wewnętrzną - formaty danych, rozmiary, itp.

Rozdział 6 - "Uwarunkowania" - oraz "Teksty programów" należy opracować zgodnie z zasadami podanymi poprzednio. Natomiast w "Przebiegi testowania" należy przedstawić wyniki testowania zarówno wszystkich pojedynczych programów, jak i grup programów problemowych oraz wyniki testowania pracy programów obsługi zbiorów.

Język problemowy

Dokumentacja techniczna oprócz wymienionych elementów stałych winna zawierać szereg elementów specyficznych wynikających z definicji języka problemowego.

Rozdział 2 - "Ogólny opis programu tłumaczącego" - należy tu przedstawić ogólne zadania tego programu, zasadę jego działania /ilustrując odpowiednim schematem/ oraz zwrócić uwagę na fakt, iż jest to formalnie pojedynczy program.

Rozdział 3 - "Struktury poszczególnych rozkazów" - opisać i zilustrować szczegółowymi schematami, budowę fragmentów programu tłumaczącego, odnoszących się kolejno do wszystkich rozkazów i wszystkich wariantów, wskazując, jakie procedury z biblioteki problemowej są wykorzystywane.

Rozdział 4 - "Opisy symboli" - opracować, dla programu tłumaczącego, tak jak dla "pojedynczego programu".

Rozdział 5 - "Uwarunkowania" - przy opracowywaniu tego rozdziału należy stosować zasady podane dla pojedynczego programu - wersja b/. Należy przedstawić pełny "tekst źródłowy" programu tłumaczącego w postaci oryginalnego wydruku.

"Wyniki testowania" - należy pokazać, jak z programu napisanego w języku problemowym program tłumaczący generuje teksty programów w języku algorytmicznym.

System problemowy

System problemowy można stworzyć, jako oprogramowanie pośrednie otwarte - wersja b/ lub jako oprogramowanie pośrednie językowe - wersja c/. Biorąc pod uwagę strukturę jest to język problemowy oraz stałe zbiory danych i ich oprogramowanie. Wobec tego dokumentacja techniczna systemu problemowego, oprócz stałych czterech pierwszych elementów powinna zawierać w wersji b/ opisy techniczne ww. części systemu. I tak:

Rozdział 2 - "Ogólna struktura systemu" - należy pokazać powiązania między oprogramowaniem problemowym zorganizowanym w postaci języka problemowego, opartego na specjalnej /własnej dla danego systemu/ bibliotece problemowej, stałymi zbiorami danych /stałych i aktualnych/ oraz oprogramowaniem obsługującym te zbiory. Strukturę systemu należy zilustrować ogólnym schematem.

Rozdział 3 - "Opis biblioteki problemowej" - opracować wg zasad podanych dla "Biblioteki".

Rozdział 4 - "Opis języka problemowego" - przedstawić według zasad podanych dla "Języka problemowego".

Rozdział 5 - "Szczegółowy opis zbiorów" - opracować według zasad dotyczących technicznego opisu zbiorów stałych, podanych dla "Małego systemu".

Rozdział 6 - Opis programów obsługi zbiorów" - każdy z programów obsługi zbiorów stałych scharakteryzować oddzielnie, w odpowiednim podrozdziale. Charakterystyka techniczna każdego z tych programów winna zawierać następujące elementy:

- funkcja programu,
- szczegółowa struktura opracowania według zasad dla pojedynczego programu,
- opis symboli, tak jak dla pojedynczego programu,
- tekst źródłowy,
- wyniki testowania.

Rozdział 7 - "Uwarunkowania" - co do formy winien być identyczny z rozdziałami charakteryzującymi inne typy oprogramowania.

Jeżeli system problemowy był pisany w ramach systemu zintegrowanego - a więc w wersji c/, to dokumentacja techniczna winna, w odpowiednich miejscach, uwzględnić ten fakt. I tak w opisie procedur problemowych, języka problemowego, struktur zbiorów i oprogramowania obsługi zbiorów stałych należy powołać się na odpowiednie mechanizmy /narzędzia/ programowe, jakie daje do dyspozycji programiście podsystemowy dany system zintegrowany. Ponadto w rozdziale 7 - "Uwarunkowania" należy wyraźnie podać, w jakim systemie zintegrowanym jest tworzony opisywany system problemowy.

Przy tworzeniu dokumentacji technicznej niezmiernie istotną sprawą jest osiągnięcie takiego poziomu szczegółowości, który umożliwiłby wykonywanie dowolnych zabiegów pielęgnacyjnych nie tylko przez autorów danego oprogramowania. Temu celowi służą opisy znaczenia poszczególnych symboli, jak również wszelkie rysunki - schematy. Stosując schematy należy przestrzegać zasady opracowywania ich według hierarchii, poczynając od ogólnych schematów struktury całego opisywanego oprogramowania. Następnie, każdy blok ze schematu bardziej ogólnego rozwijać w odręb-

ny schemat, aż do schematów będących obrazami odpowiednich fragmentów oprogramowania. Poszczególne schematy dowolnego poziomu nie powinny jednak być zbyt skomplikowane, gdyż utrudnia to jego zrozumienie. Schematy należy w prosty sposób skorelować z odpowiednimi wydrukami tekstów programów.

Ze względu na fakt, iż dokumentacja techniczna musi być bardzo obszerna, należy ściśle przestrzegać generalnej zasady /sformułowanej w [3] i [2] - równoległego opracowywania dokumentacji technicznej z innymi pracami procesu tworzenia oprogramowania.

W artykule zaproponowano stosunkowo rozwinięte i dość zróżnicowane formy dokumentowania oprogramowania dla komputerowego wspomagania projektowania. Jako pierwsza tego typu propozycja wymaga wnikliwej analizy i weryfikacji przez szerokie grono zainteresowanych. Dopiero po pewnych dyskusjach i wprowadzeniu wyników z owych dyskusji korekt, można będzie zacząć eksperymentalnie stosować owe zasady. Przestrzeganie tych zasad będzie zapewne żmudne i pracochłonne. Jest więc sprawą odrębną podjęcie prac nad komputerowym wspomaganiem dokumentowania oprogramowania. Należy też zwrócić uwagę, iż propozycje zawarte w niniejszym artykule nie wychodzą poza tradycyjne koncepcje dotyczące form dokumentowania oprogramowania użytkowego. Natomiast niektórzy autorzy [10] proponują podejście odmienne, całkowicie zrywając z owymi tradycyjnymi koncepcjami. Uwzględniając ten fakt sądzimy, iż dla nieprofesjonalnych użytkowników formy opisu zaproponowane w niniejszym artykule są właściwsze. Wiele z tych informacji można podawać użytkownikowi systemów dialogowych w formie odpowiednich komunikatów generowanych na ekranie monitora - to jednak zagadnienie jak wspomniano na wstępie nie jest przedmiotem niniejszych rozważań.

L i t e r a t u r a :

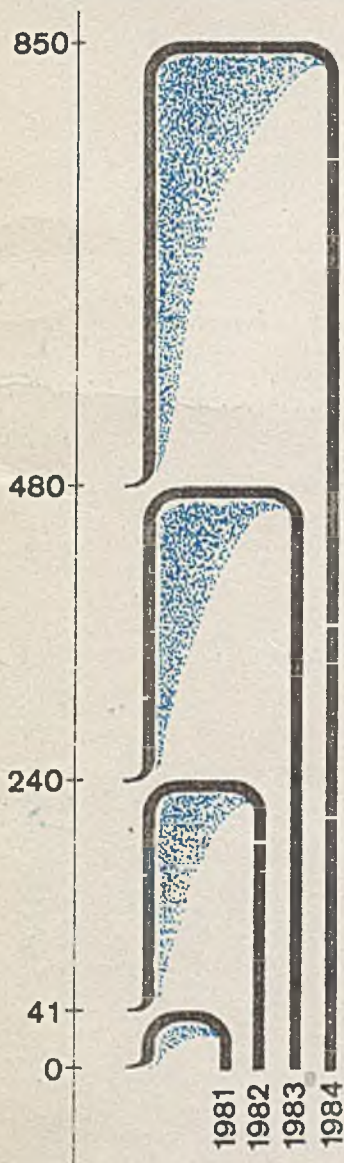
- [1] Biblioteka matematyczna PDP MATLIB.
- [2] St. Bonkowicz-Sittauer - Prowadzenie i rozpowszechnienie informacji o oprogramowaniu - PIRIO. Biuletyn Informacyjny Nauki i Techniki Komputerowej nr 1-2, 1982, IMM, Warszawa.
- [3] St. Bonkowicz-Sittauer. Dokumentacja programów użytkowych. Ref. : III Krajowy przegląd zastosowań techniki komputerowej w przemyśle maszynowym. Tom. II. Część I. Poznań 1976.
- [4] St. Bonkowicz-Sittauer. Założenia dla normy - Forma i treść dokumentacji oprogramowania dla KWP. Warszawa, 1983. Praca wykonana w IMM na zlecenie OBR-TEKOMA.
- [5] St. Bonkowicz-Sittauer. Metody realizacji typowych procesów tworzenia oprogramowania użytkowego przeznaczonego dla prac projektowania konstrukcji i technologii. Etap I. Warszawa 1983.
- [6] IBM Application Program GH20-0205-4. System/360 Scientific Subroutine Package. Version III. Programmer's Manual. Program Number 360 A-CM-03X. 1970.
- [7] Z. Kierzkowski. Modele i synteza oprogramowania systemów komputerowego wspomagania projektowania. Politechnika Poznańska, seria "Rozprawy", nr 122, Poznań 1980.
- [8] Ocena stanu Informatyki w Polsce. Sekretariat KI. Warszawa, wrzesień 1983. Materiał dla Sejmowej Komisji Nauki i Postępu Technicznego.
- [9] M. Paprzycki. Zasady atestacji informatycznych systemów automatyzacji prac inżynierskich. Ref. V Międzynarodowa Konferencja naukowo-techniczna nt. : "Komputerowe wspomaganie projektowania". Rydzyna 12 - 15 czerwca 1983.
- [10] Resortowy katalog programów obliczeń inżynierskich. OBR-Podstawy Technologii i Konstrukcji Maszyn - TEKOMA, Ministerstwo Przemysłu Maszynowego.



CENTRUM NAUKOWO-PRODUKCYJNE
SYSTEMÓW STEROWANIA

40-153 KATOWICE, ul. Armii Czerwonej 160

MERASTER -EKSPORTER SYSTEMÓW MIKROKOMPUTEROWYCH



← EKSPORT SYSTEMÓW DO
KRAJÓW SOCJALISTYCZNYCH
W LATACH 1981-1984

ZASTOSOWANIA SYSTEMÓW

- STANOWISKA AUTOMATYZACJI BADAŃ I EKSPERYMENTÓW NAUKOWYCH
- WIELOFUNKCYJNE STANOWISKA DLA OBSŁUGI PROCESU DYDAKTYCZNEGO
- STANOWISKA ZBIERANIA I PRZEKAZYWANIA DANYCH DLA PRACY AUTOMATYCZNEJ I W SYSTEMACH TELEPRZETWARZANIA
- SIECI KOMPUTEROWE

MERASTER OFERUJE:

- UŻYTKOWE SYSTEMY OPROGRAMOWANIA
- OPROGRAMOWANIE SYSTEMOWE
- USŁUGI SOFTWARE'OWE
- SERWIS

TEL. 587-206 597-086
TELEKS 031 5958
mest pl

