POLSKIE TOWARZYSTWO CYBERNETYCZNE
Oddział w GDAŃSKU

POLISH CYBERNETICAL SOCIETY
ПОЛЬСКОЕ КИБЕРНЕТИЧЕСКОЕ ОБЩЕСТВО
SOCIÉTÉ CYBERNETIQUE POLONAISE

V

KONFERENCJA UŻYTKOWNIKÓW MINIKOMPUTERA MERA-400

GDAŃSK 1988

INSTYTUT OKRETOWY
POLITECHNIKI GDANSKIEJ

POROZUMIENIE UŻYTKOWNIKOW
MINIKOMPUTERA MERA - 400
PRZY POLSKIM TOWARZYSTWIE
CYBERNETYCZNYM
ODDZIAŁ W GDANSKU

V KONFERENCJA
UŻYTKOWNIKOW MINIKOMPUTERA
MERA - 400

Materiały

Do użytku wewnętrznego

Gdańsk, 5 - 7 października 1988

KOMITET ORGANIZACYJNY KONFERENCJI

Przewodniczący:

Andrzej Braniecki, Instytut Okrętowy

Sekretarz naukowy i redaktor materiałów:
Stefan Zieliński, PTC Oddział Gdańsk, Instytut Okrętowy PG.

Sekretarz organizacyjny:
Mirosław Izdebski

Rada Programowa Konferencji - Radą Programowa Porozumienia Użytkowników Minikomputera MERA-400

1. Andrzej BRANIECKI /przewodniczący/
2. Wojciech BADURA
3. Jerzy DŻOGA
4. Tadeusz GRALL
5. Henryk KWAPINSKI
6. Krzysztof MIZIOŁEK
7. Kazimierz NOWACKI
8. Ryszard WERBINSKI
9. Jan WIERZBICKI
10. Urszula WOZNIAK
11. Stefan ZIELINSKI

SPIS TREŚCI

WSTĘP 5

OPROGRAMOWANIE UŻYTKOWE MINIKOMPUTERA MERA-400
W. Badura 7

LOKALNE SIECI KOMPUTEROWE
K. Anzelewicz 15

GENERATOR RAPORTÓW BAZY DANYCH NITRIN
A. Ziółkowski 27

SYSTEM OBSŁUGI KATALOGU I PROWADZENIA DOKUMENTACJI BIBLIOTEKI
J. Węgrzyn 41

ZESTAW PROGRAMÓW DO OBLICZANIA FUNKCJI AUTOKORELACJI I WIDMA SPEKTRALNEJ GĘSTOŚCI MOCY.
PRZYKŁADY ZASTOSOWAŃ
P. Nowak 47

PROCESY I METODY TWORZENIA SYSTEMÓW INFORMATYCZNYCH
S. Wrycza 52

EUROGRAPHICS'87 - INFORMACJA O KONGRESIE GRAFIKI KOMPUTEROWEJ
S. Bała 73

SYSTEM PROGRAMOWANIA MODULA-2 DLA MINIKOMPUTERA MERA-400 W SYSTEMIE OPERACYJNYM CROOK-5
J. Gocałek, J. Klauziński, A. Rózga 90

RÓŻNICE POMIĘDZY JĘZYKAMI MODULA-2 I PASCAL
J. Gocałek, J. Klauziński, A. Rózga 99

KOMUNIKATY I OFERTY 105

P.Z. AMEPOL
SYSTEM KOMPUTEROWY MX-16 106

P.Z. AMEPOL
System minikomputerowy MX-46
OPROGRAMOWANIE UŻYTKOWE 113

ZUT WKTiR
Informacja ofertowa 123

LISTA UCZESTNIKÒW KONFERENCJI UŻYTKOWNIKÒW NIMIKOMPUTERA MERA-400 124

LISTA CZŁONKÒW POROZUMIENIA UŻYTKOWNIKÒW MINIKOMPUTERA MERA-400 130

W S T Ę P

V Konferencja Użytkowników Minikomputera MERA-400 odbyła się w dniach 5-7 października 1987 r. w Resortowym Ośrodku Doskonalenia kadr MHiPM w Gdańsku.

Konferencja zgromadziła około 110 uczestników i obserwatorów. Na konferencji wygłoszono 9 referatów i komunikatów.

W czasie konferencji odbyły się:

- zebranie Rady Programowej Porozumienia Użytkowników Minikomputera MERA-400
- Zebranie Plenarne Porozumienia Użytkowników Minikomputera MERA-400
- Zebranie Plenarne Sekcji Zastosowań Przemysłowych.

W trakcie Zebrania Plenarnego Porozumienia przedstawione zostały i przedyskutowane następujące zagadnienia:

- sprawozdanie finansowe za okres 1986.11.01 - 1987.09.30,
- zasady weryfikacji członków Porozumienia,
- realizacja planu pracy i preliminarz wydatków na 1987 r.,
- plan pracy Porozumienia na 1988 r.

Do najważniejszych zrealizowanych zadań planowych Porozumienia można zaliczyć:

- Zorganizowanie szkolenia w zakresie podstaw użytkowania systemu operacyjnego CROOK-5 w dniach 87.08.30 - 09.04. w Sulejowie. Przewiduje się zogranizowanie następnego turnusu takiego szkolenia, a także przygotowywany jest kurs szkoleniowy dotyczący rozszerzenia wiadomości o użytkowaniu CROOK'a-5.
- Rozpoczęto edycję Zeszytów Problemowych Porozumienia - dotychczas wydano dwa Zeszyty. W przygotowaniu są zeszyty dotyczące grafiki komputerowej, sieci lokalnych oraz systemu operacyjnego CROOK-5.

- Prowadzona jest stała ewidencja oprogramowania użytkowego eksploatowanego na minikomputerze MERA-400. Po jej zakończeniu, przewiduje się wydanie informatora o takim oprogramowaniu w postaci Zeszytu Problemowego.
- Zakończono współpracę z FMiK "ERA" w zakresie części zamiennych. Doraźną pomoc w tym zakresie, a szczególnie w zakresie pakietów procesora, prowadzić będzie PZ "AMEPOL".

Do najważniejszych przedsięwzięć planowanych do realizacji w ramach Porozumienia zaliczono:

- szkolenie użytkowników w zakresie techniki języków programowania,
- szkolenie w zakresie metod tworzenia systemów informatycznych,
- szkolenie w zakresie serwisu MERY-400 i MX-16,
- homologacja kompilatora języka FORTRAN, CROOK,
- rozpowszechnienie testów dla MERY-400.

W następnej Konferencji Użytkowników Minikomputera MERA-400 uwzględniona zostanie w szerszym zakresie tematyka metod tworzenia systemów informatycznych.

Wojciech Badura
ZUT WKTiR Piotrków Tryb.

Oprogramowanie użytkowe minikomputera MERA-400

Wstęp.

Materiał niniejszy stanowi wstępną część opracowania dotyczącego inwentaryzacji oprogramowania użytkowego eksploatowanego na minikomputerze MERA-440 (lub MX-16), wykonywanego na zlecenie Porozumienia Użytkowników Minikomputera MERA-400. Opracowanie to będzie wydane przez Oddział Gdański PTC w formie Zeszytów Problemowych Porozumienia i udostępnione wszystkim członkom.

Podstawą opracowania są ankiety nadesłane przez użytkowników oraz informacje o członkach Sekcji Zastosowań Przemysłowych Porozumienia.

Charakterystyka użytkowników.

Prezentowana informacja pochodzi od 36 użytkowników minikomputera MERA-400, przy czym

- 8 jednostek to biura projektów,
- 26 jednostek to przedsiębiorstwa przemysłowe,instytuty naukowo-badawcze i inne instytucje.

System operacyjny SOM-3 i pochodne wykorzystywane są u 24 użytkowników, a system operacyjny CROOK przez 11 użytkowników. W 6 ośrodkach wykorzystywane są obydwie rodziny tych systemów operacyjnych, przy czym w części odbywa się przenoszenie oprogramowania użytkowego z systemu SOM na CROOK.Brak jest informacji o stosowanym systemie operacyjnym u 5 użytkowników - prawdopodobnie jest to system typu SOM.

Charakterystyka oprogramowania

1. Oprogramowanie wspomagania prac inżynierskich.

Oprogramowanie wspomagania prac inżynierskich wykorzystywane jest głównie w biurach projektów oraz ośrodkach badawczo-rozwojowych. Oprogramowanie to dotyczy:

- obliczeń projektowo-konstrukcyjnych w zakresie statyki płaskich ustrojów prętowych,
- wymiarowania elementów żelbetowych,
- ochrony cieplnej budynków,
- obliczeń sieci centralnego ogrzewania i wymienników ciepła,
- projektowania instalacji wentylacyjnych,
- badania stanu zanieczyszczenia środowiska,
- projektowania tras komunikacyjnych,
- obliczeń natężenia oświetlenia,
- obliczeń elektroenergetycznych,
- przygotowania taśm sterujących dla obrabiarek numerycznych,
- projektowania kolejek górskich i wyciągów linowych,
- obliczeń kół zębatych,
- kosztorysowania,
- obliczeń geodezyjnych,
- biblioteek procedur i programów matematycznych.

Odrębnym rodzajem zastosowań minikomputera MERA-400 jest wykorzystanie do automatycznego nadzoru nad przebiegiem procesu technologicznego lub badawczego. W zastosowaniach tych wykorzystywana jest współpraca minikomputera z kanałem automatyki INTELDIGIT PI.

Wykaz użytkowników oprogramowania inżynierskiego przedstawiono w Załączniku Nr 1.

2. Oprogramowanie do wspomagania zarządzania.

Oprogramowanie wykorzystywane do wspomagania zarządzania przedsiębiorstwem rozpatrywać należy w dwóch zasadniczych grupach:

- systemy powielarne, możliwe do zastosowania w każdym przedsiębiorstwie, obsługujące takie dziedziny działalności przedsiębiorstwajak ewidencja kadrowa, finansowo-kosztowa, gospodarki materiałowej, gospodarki środkami trwałymi,bądź też elementy systemu płacowego,
- systemy indywidualne, możliwe do zastosowania w konkretnym przedsiębiorstwie lub danej branży, a związane z ewidencją i rozliczaniem produkcji, sprzedaży, płac, transportu, wspomagania technologicznego,systemami informowania kierownictwa, planowania i technicznego przygotowania produkcji; rozpowszechnieniu podlegać może metodyka projektowania takich systemów

Wykaz użytkowników oprogramowania dla celów zarządzania przedstawiono w Załączniku Nr 2.

Informację o użytkownikach oprogramowania dla innych celów przedstawiono w Załączniku Nr 3.

Stosowane dla minikomputera MERA-400 oprogramowanie tworzone jest głównie w językach FORTRAN i MACROASSEMBLER, jedynie część oprogramowania stosowanego w ZZSD "Zelmer" w Rzeszowie napisana została w języku SIMBOL. Ponadto, część oprogramowania inżynierskiego pisana jest w języku BASIC.

W 10 ośrodkach wykorzystywane jest dodatkowe oprogramowanie do tworzenia systemów użytkowych typu "bazy danych".Najpowszechniej stosowanym jest fortranowskie oprogramowanie BD-83 /VITRIN/, a ponadto bazy KOBRA (OBR AM Zabrze), SIMBAD (ŻZTT Żyrardów), ADA (Metalchem Gliwice) i specjalizowany język RAL (SR Radunia).

Minikomputer MERA-400 współpracuje także z EMC ODRA-1305. Takie zastosowanie wykorzystywane jest w SR "Radunia" i Zakładach Nawozów Fosforowych w Gdańsku.

Wykaz przedsiębiorstw

1.Wojskowy Instytut Higieny i Epidemiologii w Warszawie
2.OBR Metrologii Elektrycznej LUMEL w Zielonej Górze
3.OBR Aparatury Medycznej w Zabrzu
4.ŚPKSiUP "MOSTOSTAL" w Zabrzu
5.ZZSD "PREDOM-ZELMER" w Rzeszowie
6.BZPG "STOMIL" w Bydgoszczy
7.Rybnicki Zakład Prefabrykacji w Rybniku
8.Przemysłowe Centrum Optyki w Warszawie
9.HSO "SANDOMIERZ" w Sandomierzu
10.Cz.Z.P.L. "WARTA" w Częstochowię
11.WSK "PZL-GORZYCE" w Gorzycach
12.OZOS "STOMIL" w Olsztynie
13.Żyrardowskie Zakłady Tkanin Technicznych w Żyrardowie
14.HSO "SZCZAKOWA" w Jaworznie
15.Z.A.Ch. "METALCHEM" w Gliwicach
16.Stocznia Remontowa "RADUNIA" w Gdańsku
17.BP-BBO "Miastoprojekt-Łódź" w Łodzi
18.Instytut Włókiennictwa w Łodzi
19.Instytut Niskich Temperaturi Badań Systemowych PAN we Wrocławiu
20.Biuro Projektów Budownictwa Wiejskiego w Toruniu
21.Biuro Projektów Budownictwa Komunalnego w Łodzi
22.BPBO "Miastoprojekt" w Krakowie
23.Fabryka Maszyn Górniczych "PIOMA" w Piotrkowie Tryb.
24.COBR Maszyn Włókienniczych CENARO w Łodzi
25.BP-BBO "Miastoprojekt-2" w Łodzi
26.Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne w Poznaniu
27.Biuro Projektów Budownictwa Morskiego "BIMOR" w Szczecinie
28.Gdańskie Zakłady Nawozów Fosforowych w Gdańsku
29.Komenda Wojewódzka Straży Pożarnych w Elblągu
30.Zespół Opieki Zdrowotnej w Zakopanem

31.Wojewódzki Ośrodek Informatyki w Skierniewicach

32.Zakład Informatyki Spółdzielni Mieszkaniowych w Krakowie

33.Biuro Projektów "NAFTOPROJEKT" w Warszawie

34.Wojewódzkie Biuro Projektów w Zabrzu

35.H.S.Wś "JAROSZOWIEC" w Jaroszowcu

36.Instytut ~~Maszyn~~ Maszyn Przepływowych PŁ w Łodzi

Załącznik Nr 1. Użytkownicy oprogramowania inżynierskiego

L.P.	Nazwa użytkownika	System Operacyjny	Obl. prętowe	Obl.żelbetowe	Ochrona cieplna	Sieci C.O.	Wentylacje	Ochr.środowiska	Trasy komunikac	Oświetlenie	Obl. elektryczne	Obrabiarki num.	Kolejki górskie	Koła zębate	Kosztorysowanie	Obl.geodezyjne	Obl.matematyczne	Systemy automat
1.	BPBM BIMOR Szczecin			+	+													
2.	BPBBO Miastoprojekt2 Łódź	SOM													+			
3.	BPBO Miastoprojekt Kraków	SOM	+	+		+		+		+					+			
4.	BPBK Łódź	SOM	+			+		+							+			
5.	BPBW Toruń	SOM				+									+			
6.	BPBBO Miastoprojekt Łódź		+	+	+	+	+	+	+	+	+						+	
7.	BP Naftoprojekt Warszawa	SOM	+		+	+		+		+								
8.	WBP Zabrze	SOM	+	+		+			+	+	+				+			
9.	FMG PIOMA Piotrków Tryb.	SOM										+						
10.	COBR MM GENARO Łódź	SOM												+				
11.	OPC-K Poznań	SOM														+		
12.	ŚPKSiUP Mostostal Zabrze	SOM											+					
13.	BZPG STOMIL Bydgoszcz	SOM																+
14.	OZOS Stomil Olsztyn	SOM																+
15.	IMP PŁ Łódź	SOM																+
16.	WIHiE Warszawa	SOM															+	
17.	INTiBS PAN Wrocław	SOM															+	

Załącznik nr 2.Użytkownicy oprogramowania do celów zarządzania

L.P.	Nazwa przedsiębiorstwa	System Operacyjny	Fin.-Kosztowy	Środki Trwałe	Kadry	Płace	Rozl. produkcji	Rozl. sprzedaży	Wyroby Gotowe	Gosp. Materiałowa	Wspomag.technolog.	Analizy i Planow.	T P P	Rozl. transportu	S I K	
1.	Z-dy Nawozów Fosforowych Gdańsk	CROOK	+	+												
2.	OPG-K Poznań			+			+	+								
3.	OBR ME LUMEL Zielona Góra	CROOK				+	+									
4.	OBR Ap.Medycznej Zabrze	SOM			+					+						
5.	ŚPKSiUP Mostostal Zabrze	SOM	+		+	+	+	+		+		+				
6.	ZZSD Predom-Zelmer Rzeszów	SOM	+				+				-	+	+			
7.	PCO Warszawa				+	+							+			
8.	HSO Sandomierz	SOM		+	+	+	+	+	+	+	+				+	
9.	CzZPL Warta Częstochowa	SOM	+	+		+	+	+		+		+				
10.	WSK PZL-GORZYCE Gorzyce	SOM			+	+	+	+				+	+			
11.	ŻZTTechn. Żyrardów	SOM				+				+				+		
12.	HSO Szczakowa Jaworzno	CROOK		+	+	+	+	+	+		+				+	
13.	ZACh Metalchem Gliwice	CROOK	+	+		+	+			+						
14.	SR Radunia Gdańsk	CROOK	+	+			+	+		+					+	
15.	WOI Skierniewice	SOM				+				+						
16.	ZI Sp.Mieszkaniowych Kraków	CROOK								+				+		
17.	HSW Jarocszowiec	SOM				+	+	+	+		+			+		
18.	BPBO Miastoprojekt Kraków	SOM				+		+								
19.	BPBK Łódź	SOM						+								
20.	WBP Zabrze	SOM				+										
21.	INTiBS PAN	SOM		+		+										
22.	ZOZ Zakopane	CROOK			+	+				+						
23.	WIHiE Warszawa	SOM			+	+										

Załącznik nr 3. Systemy pozostałe

1.	ZOZ Zakopane	CROOK	Statystyka izby przyjęć i Pogotowia Ratunkowego
2.	WOI SKIERNIEWICE	SOM	naliczanie czynszów
			sprawozdawczość budżetowa
			pośredictwo pracy
			rozliczenie mandatów kredytowych
			rozliczenia funduszy miejskich
3.	Instytut Włókiennictwa Łódź	SOM	ewidencja prac badawczych
4.	KWSP Elbląg	SOM	informacja o zagrożeniach p-pożar.
5.	ZI Spółdz.Mieszk. Kraków	CROOK	rozliczanie czynszów

Krzysztof Anzelewicz
Politechnika Gdańska
Instytut Okrętowy

Lokalne Sieci Komputerowe

Wstęp.

Lokalne Sieci Komputerowe (LSK) (*ang. Local Area Networks -LAN*) powstały w połowie lat siedemdziesiątych w USA. Prace konstrukcyjne związane z utworzeniem LSK podjęte zostały już w 1973 roku przez firmę Xerox. Skonstruowana przez tę firmę sieć otrzymała nazwę ETHERNET. Z kolei w roku 1976 firma Datapoint utworzyła inną sieć lokalną, której nadano nazwę ARCnet. Dopiero jednak rozwój technologi VLSI na przełomie lat siedemdziesiątych i osiemdziesiątych umożliwił szerokie rozpowszechnienie LSK w związku ze znacznym obniżeniem kosztów sprzętu niezbędnego do tworzenia sieci oraz z pojawieniem się mikrkomputerów typu IBM PC.

Lokalną sieć komputerową stanowi zespół co najmniej kilku połączonych systemów mikrokomputerowych rozmieszczonych na niewielkim obszarze, wraz z zestawem środków sprzętowych i programowych, które umożliwiają tym mikrokomputerom wzajemną komunikację oraz korzystanie ze wspólnych zasobów.

Mikrokomputery komunikują się ze sobą poprzez podsystem transmisji o następujących parametrach:

- tani ośrodek transmisji (skrętka telefoniczna, kabel współosiowy, światłowód)
- sprzężenie mikrokomputerów z ośrodkiem transmisji za pomocą wyspecjalizowanych sterowników LSK,
- transmisja szeregowa z szybkością od 1 do 10 Mb/s,
- niezależność szybkości transmisji od szybkości pracy mikrokomputerów dołączonych do sieci,

- możliwość komunikacji dowolnej pary mikrokomputerów dołączonych do sieci,
- możliwość komunikacji mikrokomputerów w trybie pełnego lub selektywnego rozgłaszania (*ang. broadcast, multicast-group addressing*),
- stosowanie zdecentralizowanej procedury dostępu do ośrodka transmisji.

Wymienione cechy podsystemu transmisji LSK sprawiają, iż dostarcza on nowych możliwości systemom teleinformatycznym. W szczególności umożliwia on:

- efektywne wykorzystanie zasobów sieciowych dzięki prostocie i niezawodności komunikacji pomiędzy wszystkimi mikrokomputerami,
- łatwą rozbudowę sieci,
- uzyskanie wysokiego stopnia integracji przetwarzania i przechowywania informacji, co wyraża się możliwością organizowania przetwarzania wielokomputerowego oraz budową rozproszonych baz danych.

Konfiguracje sieci.

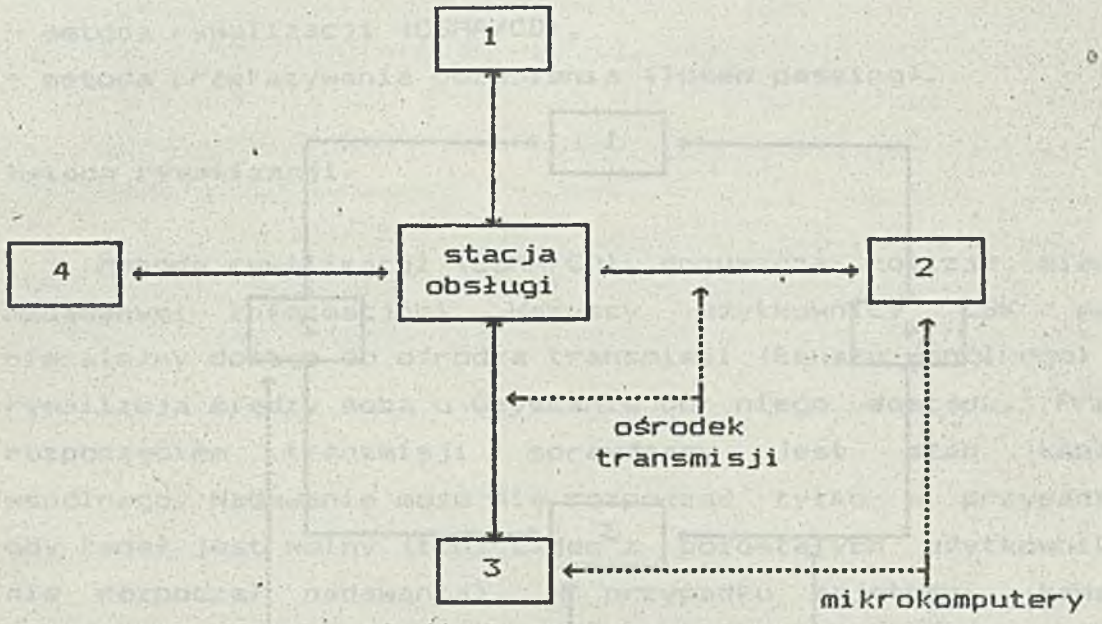
Stosowane są trzy podstawowe konfiguracje sieci:

- gwiaździsta (star),
- magistralowa (bus),
- pętlowa (ring).

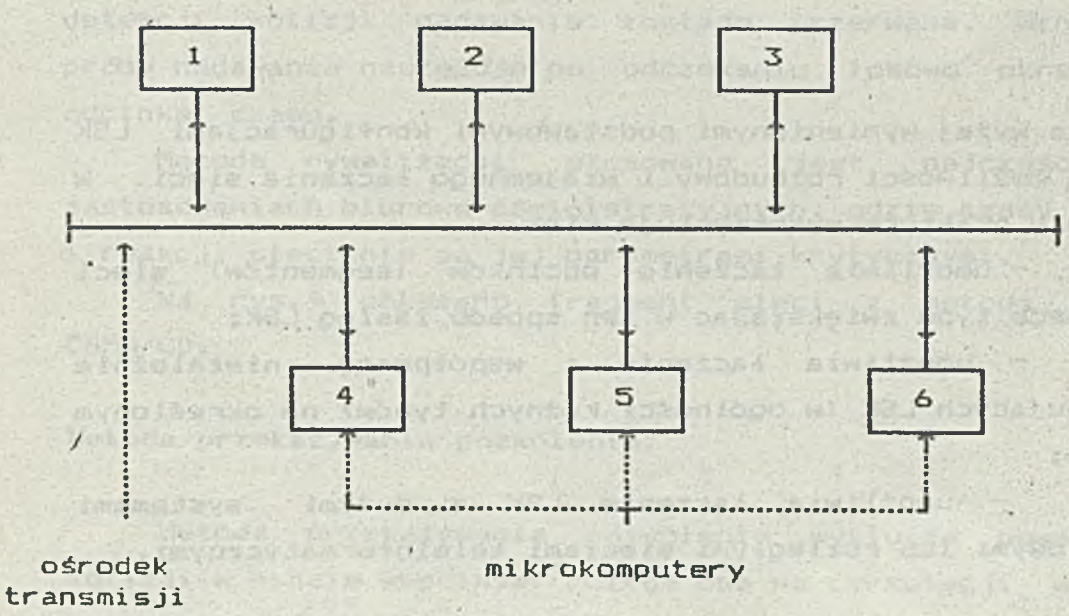
W przypadku pierwszej z wymienionych konfiguracji mikrokomputery użytkowników dołączone są do centralnego mikrokomputera - stacji obsługi, który między innymi pośredniczy w komunikacji między poszczególnymi mikrokomputerami sieci. Konfigurację taką przedstawia rys.1.

Sieć o konfiguracji magistralowej charakteryzuje się tym, że wszystkie mikrokomputery tej sieci komunikują się poprzez wspólny ośrodek transmisji mający charakter magistrali. Przesyłana w takiej sieci informacja dociera

praktycznie jednocześnie do wszystkich mikrokomputerów (w rzeczywistości występują opóźnienia związane z propagacją sygnałów w ośrodku transmisji). Konfigurację magistralową przedstawia rys.2.

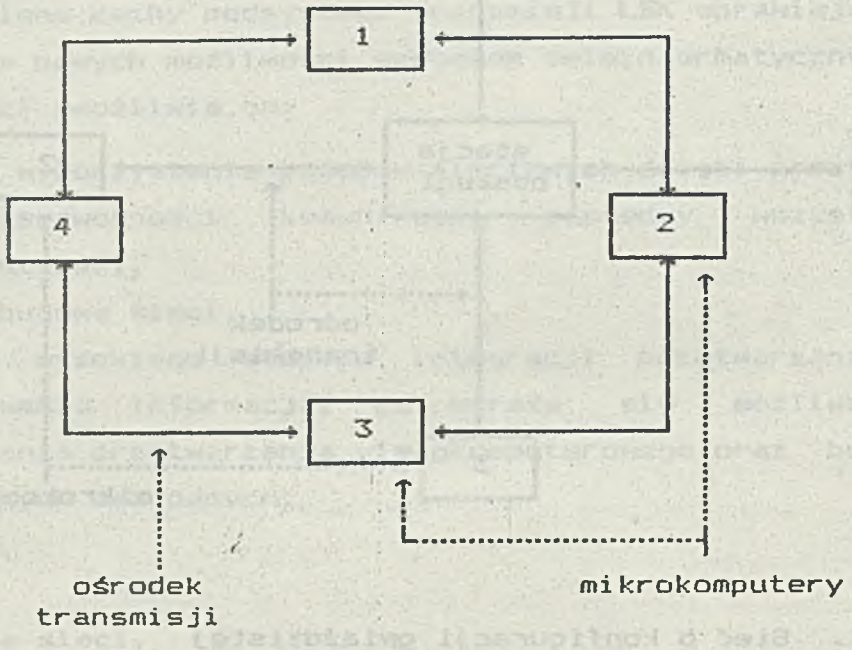


Rys.1. Sieć o konfiguracji gwiaździstej



Rys.2. Konfiguracja magistralowa sieci

Konfiguracja pętlowa charakteryzuje się tym, że mikrokomputery LSK połączone są odcinkami ośrodka transmisji tworząc w efekcie zamkniętą pętlę. Informacja wysyłana z jednego węzła (mikrokompuitera) jest retransmitowana przez kolejne węzły aż do powrotu do węzła nadającego. Konfigurację pętlową przedstawia rys.3.



Rys.3. Konfiguracja pętlowa sieci

Poza wyżej wymienionymi podstawowymi konfiguracjami LSK istnieją możliwości rozbudowy i wzajemnego łączenia sieci. W tym celu w praktyce wykorzystuje się:

Repeater - umożliwia łączenie odcinków (segmentów) sieci tego samego typu zwiększając w ten sposób zasięg LSK;

Bridge - umożliwia łączenie i współpracę niezależnie funkcjonujących LSK (w ogólności różnych typów) na określonym obszarze;

Gateway - umożliwia łączenie LSK z dużymi systemami komputerowymi lub rozległymi sieciami teleinformatycznymi.

Metody dostępu do ośrodka transmisji.

W związku z tym, że w LSK o konfiguracji pętlowej lub

magistralowej stosowany jest wspólny dla wszystkich mikrokomputerów ośrodek transmisji, konieczne jest stosowanie metod (procedur) dostępu do tego ośrodka, zapewniających poprawną pracę sieci.

Podstawowe są dwie następujące metody dostępu:

- metoda rywalizacji (CSMA/CD),
- metoda przekazywania pozwolenia (Token passing).

Metoda rywalizacji.

Metoda rywalizacji (CSMA/CD) dopuszcza kolizje między nadawanymi informacjami. Wszyscy użytkownicy LSK mają niezależny dostęp do ośrodka transmisji (kanału wspólnego) i rywalizują między sobą o uzyskanie do niego dostępu. Przed rozpoczęciem transmisji sprawdzany jest stan kanału wspólnego. Nadawanie może się rozpocząć tylko w przypadku, gdy kanał jest wolny (tzn. żaden z pozostałych użytkowników nie rozpoczął nadawania). W przypadku zajętości kanału nadawanie informacji jest wstrzymywane do chwili jego zwolnienia. W trakcie nadawania stan kanału jest kontrolowany pod kątem wystąpienia ewentualnej kolizji (kolizja może wystąpić w sytuacji, gdy nadawanie rozpoczyna jednocześnie co najmniej dwóch użytkowników - jest to możliwe ze względu na występowanie opóźnień propagacyjnych w kanale). W przypadku detekcji kolizji nadawanie zostaje przerwane. Wznowienie próby nadawania następuje po odczekaniu losowo określonego odcinka czasu.

Metoda rywalizacji stosowana jest najczęściej w zastosowaniach biurowo-administracyjnych, gdzie czasy dostępu i reakcji sieci nie są jej parametrami krytycznymi.

Na rys.4 pokazano fragment sieci z metodą dostępu CSMA/CD.

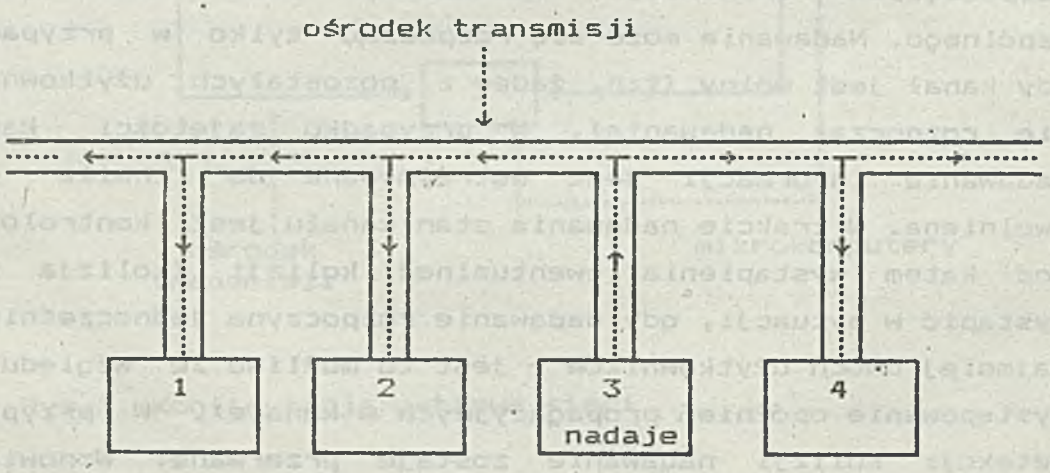
Metoda przekazywania pozwolenia.

Metoda przekazywania pozwolenia wyklucza powstawanie kolizji w kanale wspólnym. Polega ona na cyrkulacji w sieci specjalnej ramki, zawierającej między innymi tzw. bit pozwolenia (Token). Użytkownik uzyskuje możliwość dostępu do kanału (tzn. wykorzystania wolnej ramki) bezpośrednio po odbiorze bitu pozwolenia. Wykorzystywana ramka jest zwalniana

po jej powrocie do nadawcy (tzn. ustawiany jest ponownie bit pozwolenia). W przypadku braku informacji gotowej do nadawania, ramka zawierająca bit pozwolenia zostaje przekazana kolejnemu użytkownikowi.

Metoda przekazywania pozwolenia jest metodą deterministyczną ponieważ znając parametry LSK można obliczyć chwilę, w której dany użytkownik uzyska dostęp do kanału wspólnego. LSK wykorzystujące tę metodę dostępu znajdują szerokie zastosowanie w zakładach przemysłowych przy sterowaniu procesami produkcyjnymi. Dodatkową cechą tej metody dostępu jest łatwość wprowadzenia mechanizmu priorytetów dla poszczególnych użytkowników sieci.

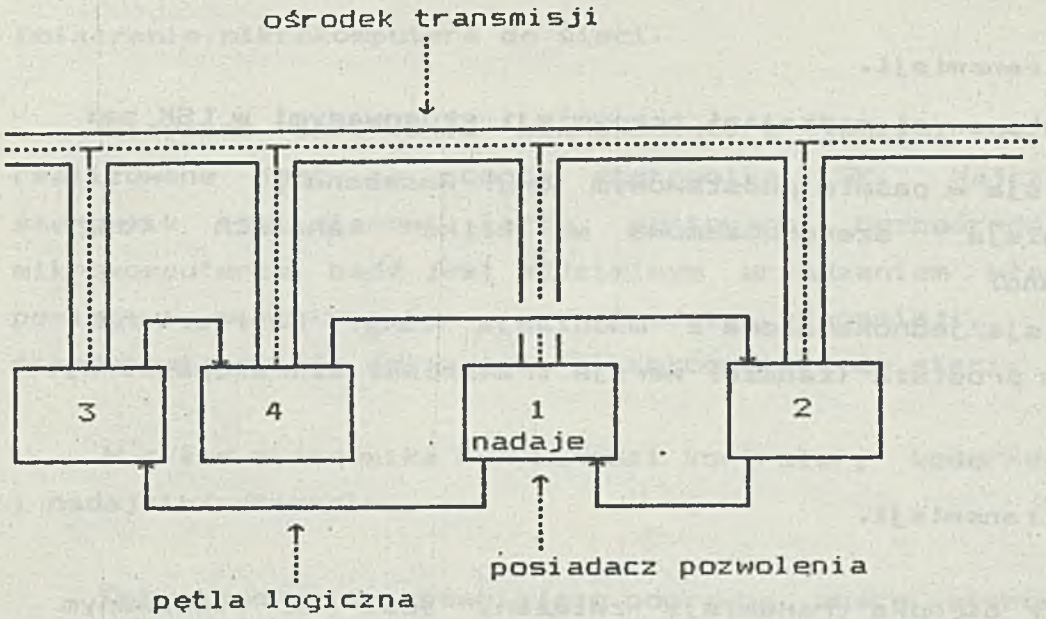
Wyżej wymieniona metoda dostępu znajduje zastosowanie zarówno w sieciach o konfiguracji magistralowej jak i pętlowej.



Rys.4. LSK z metodą rywalizacji.

Przekazywanie pozwolenia w sieci magistralowej.

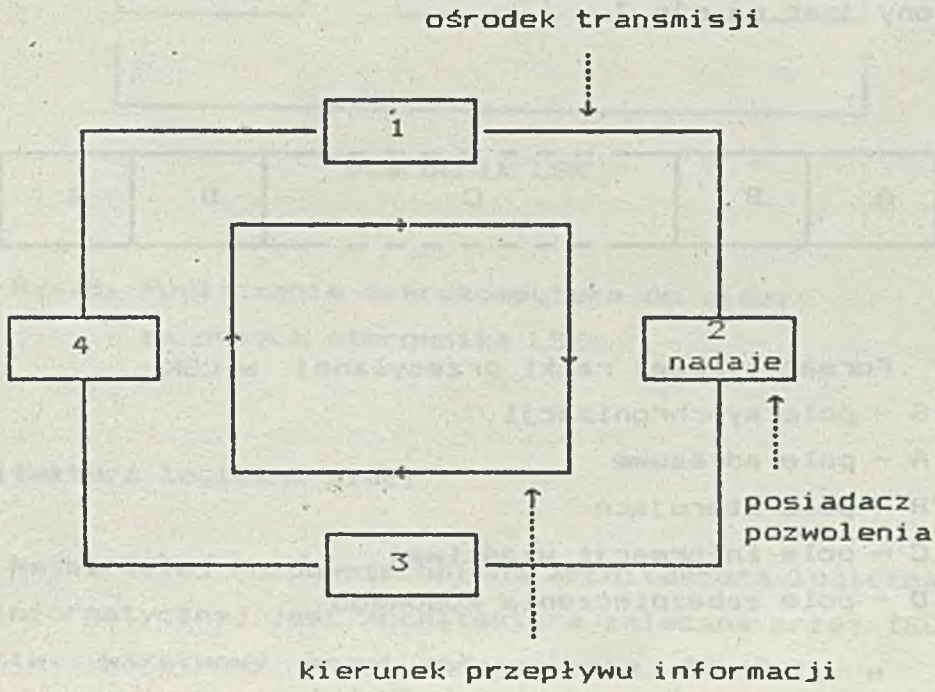
W sieci o konfiguracji magistralowej (Token bus) ramka z bitem pozwolenia przesyłana jest w tzw. pętli logicznej wynikającej z następstwa adresów poszczególnych mikrokomputerów pracujących w sieci. Adresy te nadawane są mikrokomputerom w trakcie inicjalizacji LSK. Każdy z mikrokomputerów zna adres swojego poprzednika i następcy. Sposób funkcjonowania takiej sieci ilustruje rys.5.



Rys.5. Sieć magistralowa z przekazywaniem pozwolenia

Przekazywanie pozwolenia w sieci pętlowej.

W sieci o konfiguracji pętlowej (Token ring) ramka z bitem pozwolenia przesyłana jest zgodnie z kolejnością określoną przez połączenia pomiędzy mikrokomputerami. Sytuację tę ilustruje rys.6.



Rys.6. Sieć pętlowa z przekazywaniem pozwolenia

Rodzaje transmisji.

Podstawowymi rodzajami transmisji stosowanymi w LSK są:

- Transmisja w paśmie podstawowym (*ang. Baseband*),
- Transmisja szerokopasmowa w kilku kanałach (*ang. Broadband*)
- Transmisja jednokanałowa z modulacją (*ang. Carrierband*). Jest to prostsza (tańsza) wersja transmisji szerokopasmowej.

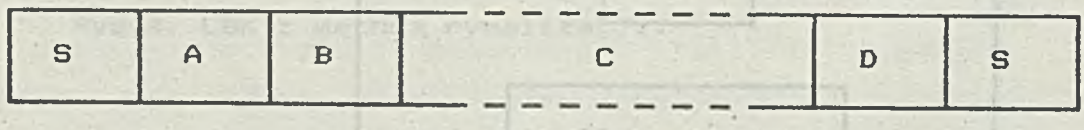
Ośrodek transmisji.

Dobór ośrodka transmisji związany jest ze stosowanym rodzajem sieci. Jako ośrodek transmisji wykorzystuje się:

- skrętkę telefoniczną,
- kabel koncentryczny,
- kabel światłowodowy.

Przesyłanie informacji.

Informacje w LSK przesyłane są w postaci tzw. ramek. Istnieją różne formaty ramek stosowanych w LSK, jednakże w każdym przypadku posiadają one określoną długość minimalną i maksymalną. Format typowej ramki przesyłanej w LSK przedstawiony jest na rys.7.



Rys.7 Format typowej ramki przesyłanej w LSK

S - pole synchronizacji

A - pole adresowe

B - pole sterujące

C - pole informacji właściwej

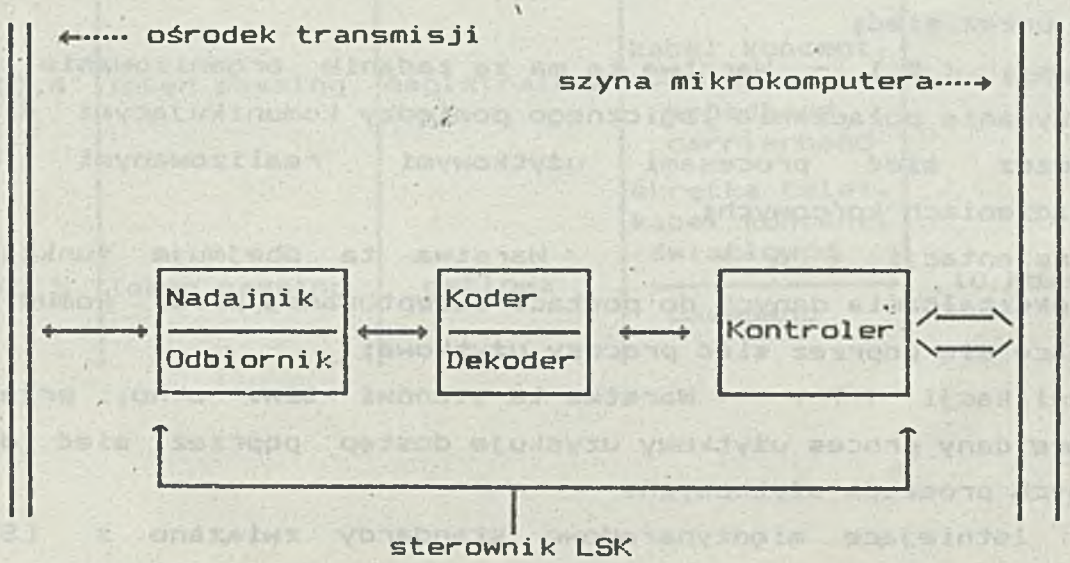
D - pole zabezpieczenia kodowego

Dołączenie mikrokomputera do sieci.

Fizyczne dołączenie mikrokomputera (węzła) do sieci realizowane jest za pomocą sterownika LSK. Najczęściej sterownik ten stanowi kartę montowaną bezpośrednio w mikrokomputerze, bądź jest oddzielnym urządzeniem włączanym pomiędzy mikrokomputer a ośrodek transmisji. Rys.8 przedstawia sposób dołączenia mikrokomputera do sieci.

W skład sterownika LSK wchodzi kontroler, koder/dekoder i nadajnik/odbiornik.

Oprogramowanie zapewniające poprawną pracę sterownika, jak też kontrolę pracy LSK, przechowywane jest w pamięci ROM sterownika.



Rys.8. Podłączenie mikrokomputera do sieci za pomocą sterownika LSK

Architektura logiczna sieci

Najbardziej rozpowszechnioną architekturą logiczną sieci teleinformatycznej jest architektura zalecana przez ISO. Jest to tzw. warstwowy model odniesienia ISO/OSI, w którym wyróżnia się siedem warstw z przyporządkowanymi im odpowiednimi funkcjami komunikacyjnymi. Są to warstwy :

- Fizyczna (1) - Warstwa ta określa elektryczne, mechaniczne oraz funkcjonalne i proceduralne charakterystyki niezbędne do uaktywnienia, utrzymania i anulowania połączeń fizycznych do transmisji cyfrowej (łącze danych) pomiędzy jednostkami sieci (stacje, węzły sieci);
- Łącza danych (2) - Warstwa ta obejmuje procedury sterowania łączem danych i realizuje funkcje ustanawiania, utrzymania i zwalniania połączeń logicznych pomiędzy jednostkami sieci za pośrednictwem tego łącza;
- Sieciowa (3) - Warstwa ta obejmuje funkcje przesyłania danych w sieci komunikacyjnej pomiędzy węzłem źródłowym i węzłem docelowym (realizowane są tu m.in. funkcje doboru trasy, sterowania przepływem strumieni danych itd.);
- Transportowa (4) - Warstwa ta obejmuje funkcje związane z przygotowaniem danych do przesłania ich przez sieć komunikacyjną oraz funkcje przesyłania danych pomiędzy urządzeniami końcowymi (komputery, końcówki) komunikującymi się przez sieć;
- Sesji (5) - Warstwa ta ma za zadanie organizowanie i utrzymanie połączenia logicznego pomiędzy komunikującymi się poprzez sieć procesami użytkowymi realizowanymi w urządzeniach końcowych;
- Prezentacji (6) - Warstwa ta obejmuje funkcje przekształcania danych do postaci akceptowanej przez komunikujące się poprzez sieć procesy użytkowe;
- Aplikacji (7) - Warstwa ta stanowi tzw. okno, przez które dany proces użytkowy uzyskuje dostęp poprzez sieć do innych procesów użytkowych.

Istniejące międzynarodowe standardy związane z LSK dotyczą w istocie dwóch pierwszych warstw modelu ISO/OSI. Ze względu na specyfikę LSK warstwa łącza danych została podzielona na dwie podwarstwy: podwarstwę dostępu do ośrodka transmisji (*ang. Media Access Control - MAC*) oraz podwarstwę sterowania łączem (*ang. Logical Link Control - LLC*). Standardy te scharakteryzowane są w następnym punkcie.

Standardy IEEE 802.n.

W połowie lat osiemdziesiątych opracowana została przez IEEE rodzina standardów 802.n dotyczących LSK w zakresie dwóch najniższych warstw sieciowej architektury logicznej.

W odniesieniu do warstwy fizycznej oraz podwarstwy MAC opracowano następujące standardy:

- 802.3 - dla sieci magistralowej z rywalizacją,
- 802.4 - dla sieci magistralowej z przekazywaniem pozwolenia,
- 802.5 - dla sieci pętlowej z przekazywaniem pozwolenia.

Tabela I charakteryzuje wyżej wymienione standardy.

Tabela I

Stand.	Metoda dostępu	Konfiguracja	Ośrodek trans. / Rodzaj trans.	Prędkość Transmisji
802.3	CSMA/CD	magistralowa	kabel koncent. / baseband	10 Mb/s
802.4	Token passing	magistralowa	kabel koncent. / broadband, carrierband	5,10 Mb/s
802.5	Token passing	pętlowa	skrętka telef. kabel koncent. światłowód / baseband	10 Mb/s

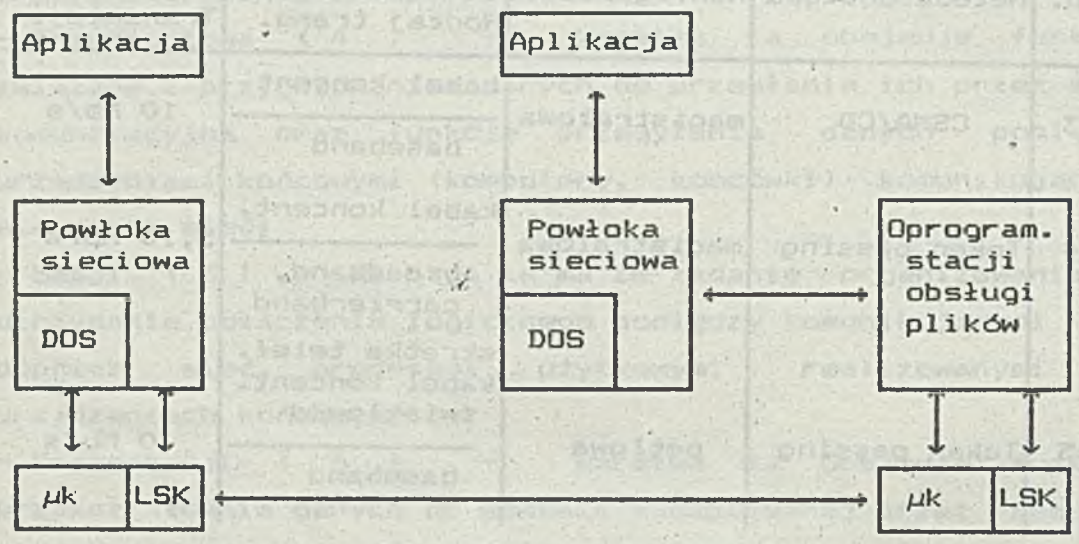
Oprogramowanie sieciowe.

Najbardziej rozpowszechnione LSK oparte są na mikrokomputerach IBM PC i kompatybilnych. Mikrokomputery te wyposażone są w system operacyjny DOS (v.3.10 i wyższe) zawierający mechanizmy zapewniające kontrolę użytkowania i dostępu do zasobów, które w środowisku sieciowym mogą być współużytkowane jednocześnie przez wielu użytkowników. Ponadto lokalny system operacyjny DOS uzupełniony jest tzw. powłoką sieciową (*ang. network shell*), stanowiącą zasadnicze oprogramowanie sieciowe. Jednym z podstawowych zadań powłoki sieciowej jest kwalifikacja odwołań programów użytkowych na

lokalne i sieciowe. Całość nadzorujesieciowy system operacyjny umieszczony w stacji obsługi.

Koncepcja oprogramowania sieciowego (sieciowego systemu operacyjnego) przedstawiaiona jest poglądowo na rys.9.

Na rysunku tym przedstawiono mikrokomputer użytkownika sieci (nie udostępniającego sieci własnych zasobów) oraz mikrokomputer będący stacją obsługi plików stanowiących zasoby sieciowe.



Rys.9.. Oprogramowanie sieciowe

W odniesieniu do modelu ISO/OSI, sieciowy system operacyjny realizuje funkcje warstw wyższych od warstwy trzeciej począwszy. Funkcje dwóch najniższych warstw tego modelu realizowane są przez sterownik LSK.

Andrzej Ziółkowski
Instytut Badań Systemowych PAN

GENERATOR RAPORTOW BAZY DANYCH VITRIN

1. Wprowadzenie.

Generator raportów RAP jest interpreterem specjalizowanego języka przeznaczonego do tworzenia wydruków o dowolnej postaci na podstawie danych przechowywanych w zbiorach bazy danych VITRIN. Dostępny w bazie danych VITRIN , program DRUK umożliwia sporządzanie raportów na podstawie danych zawartych w jednym zbiorze (ewentualnie z wykorzystaniem zbioru pomocniczego do wyszukiwania rekordów) przy czym raporty te mają ściśle określoną postać. Pisanie programów drukowania raportów w FORTRANIE jest bardzo uciążliwe. Program RAP umożliwia sporządzanie raportów na podstawie informacji zawartej w wielu zbiorach nie stwarzając ograniczeń na postać wydruków. Pisanie i późniejsze modyfikowanie programów drukujących raporty jest bardzo proste. Język umożliwia także realizację złożonego przetwarzania danych i dlatego może również zastąpić program LACZ używany w bazie danych VITRIN.

Program w języku interpretowanym przez RAP składa się z prostych instrukcji wykonywanych w naturalnej kolejności. Instrukcja gdy umożliwia warunkowe wykonywanie pewnych części programu a instrukcja dopoki wielokrotne powtarzanie.

Omawiając budowę instrukcji będziemy korzystali z następujących oznaczeń:

[ ] - nawiasy kwadratowe służą do oznaczenia części opcjonalnej instrukcji,

[¦] - kreski pionowe służą do oznaczenia alternatyw (jedna z części musi wystąpić w instrukcji),

[¦]... - nawiasy kwadratowe z kropkami służą do oznaczenia części instrukcji, która może wystąpić wielokrotnie (część po znaku ¦ występuje o ile są dalsze powtórzenia).

< > - nawiasy ostre służą do oznaczenia parametru formalnego.

2. Instrukcje dotyczące zbiorów.

otworz <zb>,<sek> [:<nzb>]

Instukcja otworz powoduje odszukanie i otwarcie zbioru o podanej nazwie <zb> na sekcji <sek> . Jednocześnie można otworzyć wiele zbiorów. Ograniczenie ilości jednocześnie otwartych zbiorów wynika jedynie z wielkości dostępnej pamięci. Opcjonalny parametr <nzb> - nowa nazwa zbioru, służy do zdefiniowania nazwy zbioru, która będzie używana w odwołaniach do zbioru w dalszej części programu. Zbiory otwarte jednocześnie muszą mieć różne <nzb>. Pominięcie tego parametru oznacza, że przyjęto <nzb> równe <zb>. W przypadku, gdy zbiory otwierane na różnych sekcjach mają te same nazwy, <nzb> przynajmniej dla jednego zbioru musi być podane. Po otwarciu zbioru dostępne są wszystkie dane występujące w dokumencie (rekordzie).

czytaj <nzb>

Instrukcja wczytania następnego dokumentu.

czytaj <nzb> : <wyr.tekstowe>

Instrukcja wczytania dokumentu o podanym identyfikatorze, który definiuje się przez podanie wyrażenia tekstowego. Sposób zapi

wyrażeń tekstowych określono w punkcie 4.

czytaj <nzb> = <wyr.arytm.>

Instrukcja wczytania dokumentu o podanym numerze fizycznym. Sposób zapisu wyrażeń arytmetycznych określono w punkcie 4.

pisz <nzb>

Instrukcja pisania do kolejnego dokumentu. Instrukcja nie zmienia identyfikatora. Osiągnięcie końca zbioru sygnalizowane jest jako błąd (zmienna $bl różna od zera).

pisz <nzb> : <wyr.tekstowe>

Instrukcja pisania dokumentu o podanym identyfikatorze. Jeżeli dokument jeszcze nie istnieje to zostanie utworzony.

pisz <nzb> = <wyr.arytm.>

Instrukcja pisania do dokumentu o podanym numerze fizycznym. Jeżeli dokument o tym numerze jeszcze nie istnieje, sygnalizowany jest błąd (zmienna $bl różna od zera).

usun <nzb>

Instrukcja oznaczania kolejnego dokumentu jako usuniętego.

usun <nzb> : <wyr.tekstowe>

Instrukcja oznaczania dokumentu o podanym identyfikatorze jako usuniętego.

usun <nzb> = <wyr.arytm.>

Instrukcja oznaczania dokumentu o pcdanym numerze fizycznym jako usuniętego. Podczas usuwania sygnalizowany jest błąd gdy usuwany dokument nie istnieje.

Przykłady:

czytaj z

```
pisz  z : " 1" + k2 + k3'3..5
usun  z = $a + 1
```

3. Dane, nazwy danych, parametry i zmienne.

Przy definiowaniu raportu lub algorytmu przetwarzania można odwoływać się do następujących wielkości:

- danych z dokumentów,
- parametrów ogólnych,
- parametrów zbioru,
- zmiennych pomocniczych.

Odwołanie do danej z dokumentu ma postać:

<dana> [.<nzb>]

gdzie <dana> jest kodem danej a <nzb> nazwą zbioru zdefiniowaną instrukcją otworz. Nazwa zbioru wraz z poprzedzającą kropką jest częścią opcjonalną. Gdy zostanie pominięta przyjmowana jest domyślnie nazwa zbioru z ostatniego odwołnia. Ponieważ można otworzyć wiele zbiorów, w których mogą występować dane o tych samych nazwach, podawanie nazwy zbioru w odwołaniu jest w ogólnym przypadku konieczne.

Odwołanie do pełnej 28-mio znakowej nazwy danej ma postać:

@<dana> [.<nzb>]

Wszystkie odwołania do parametrów oraz zmiennych rozpoczynają się znakiem $, po którym podawana jest nazwa parametru. Wyróżniono następujące parametry ogólne:

$bl - numer błędu. Zero oznacza poprawne wykonanie ostaniej instrukcji.

$ad - aktualny dzień (1..31)

$am - aktualny miesiąc (1..12)

$ar - aktualny rok (1987..)

$ns - numer strony

$nw - numer wiersza

Odwołania do parametrów zbioru mają postać:

$<par> [.<nzb>]

gdzie <par> jest nazwą parametru a <nzb> podawaną opcjonalnie nazwą zbioru. W przypadku pominięcia nazwy zbioru Przyjmuje się domyślnie, że odwołanie dotyczy zbioru do którego się ostatnio odwoływano. Można odwoływać się do wszystkich parametrów zbioru, używając nazw podanych w dokumentacji systemu VITRIN np. $ald, $dklu, $ngen. Ponadto dla każdego zbioru wprowadzono dodatkowo parametr:

$nr - aktualny numer dokumentu. Parametr ten przyjmuje wartość zero gdy w ostatnio wykonywanej instrukcji czytania, pisania lub usuwania dokumentu wystąpił błąd.

Do wykonywania obliczeń i przechowywania informacji można używać zmiennych o następujących nazwach:

Zmienne całkowite:

$a

$b

...

$z

Zmienne rzeczywiste:

$aa

$bb

...

$zz

Zmienne tekstowe:

$aaa

$bbb

...

$zzz

4. Instrukcje podstawienia i kopiowania.

Instrukcja podstawienia ma identyczną postać jak w języku FORTRAN. Po lewej stronie znaku podstawienia = może wystąpić dana z dokumentu, parametr lub zmienna, po prawej wyrażenie arytmetyczne, w którym oprócz danych i zmiennych mogą występować parametry oraz stałe liczbowe. Do zpisywania wyrażeń arytmetycznych używa się operatorów działań:

+ - dodawanie,

- - odejmowanie,

* - mnożenie,

/ - dzielenie,

oraz nawiasów okrągłych (i).

Po lewej stronie znaku podstawienia wystąpić również zmienna tekstowa. Bezpośrednio po nazwie zmiennej tekstowej musi być podany format, wg którego obliczona wartość wyrażenia arytmetycznego zostanie zapisana w tej zmiennej. Sposób zapisu formatu przedstawiono w punkcie 7.

Oprócz instrukcji podstawienia wprowadzono instrukcję kopiowania wartości tekstowych, którą zapisuje się podając dan lub zmienną tekstową, na którą tekst będzie kopiowany, a po znaku : wyrażenie tekstowe, które ma być skopiowane. W wyrażeniu mogą wystąpić następujące elementy:

@<dana> [.<nzb>] - 28-mio znakowa nazwa danej,

$<par> [.<nzb>] - parametr znakowy np. nazwa zbioru,

"<tekst>" - stała tekstowa,

$<zmienna> - zmienna tekstowa, np. $aaa, $ttt

<dana> [.<nzb>] - dana (traktowana jako tekst)

Operator konkatenacji + umożliwia łączenie tekstów w jedną całość. Ponadto dla każdego elementu wyrażenia tekstowego można określić początkowe i końcowe numery kopiowanych znakow podając odpowiednie wartości liczbowe lub nazwy zmiennych całkowitych. Ogólna postać wyrażenia tekstowego jest następująca:

[<element> ['<poczatek> .. <koniec>]] ¦ +]...

Przykłady:

$a = $a + 1

Zwiększenie wartości zmiennej całkowitej $a o 1.

wyn.wy=(war.zb-$aa)/(war+$aa)

Obliczenie wartości wyn w zbiorze wy określonej za pomocą wyrażenia arytmetycznego na podstawie wartości danej war ze zbioru zb oraz zmiennej rzeczywistej $aa.

$aaa : "Odbiorca:" + nodb.zb'6

Kopiowanie do zmiennej tekstowej $aaa tekstu "Odbiorca:" oraz

bezpośrednio po nim tekstu z danej **nodb** ze zbioru **zb** od 6-tego znaku. (Uwaga - długość tekstu zmiennej nie może przekraczać 160 znaków).

5. Gdy, dopóki.

Instrukcja **gdy** umożliwia warunkowe wykonywanie innych instrukcji. Ogólna postać tej instrukcji jest następująca:

gdy <wyrazenie logiczne> : <instrukcja>

W wyrażeniu logicznym mogą wysrępować dane, parametry, zmienne i stałe. Do zapisywania wyrażeń arytmetycznych używa się operatorów:

& - iloczyn logiczny,

¦ - suma logiczna,

> - większy,

= - równy,

- różny,

< - mniejszy,

oraz nawiasów okrągłych (i).

Relacje większy, równy, różny, i mniejszy odnoszą się do wielkości całkowitych, rzeczywistych i tekstowych.

Występująca po znaku : instrukcja może być zastąpiona blokiem (sekwencją) instrukcji ograniczonym nawiasami { i } (lub << i >>).

Przykłady:

```
gdy wyn.zb < 0 : wyn.zb = 0
gdy wyn.zb = 0 & id # "###" :
 {
 czytaj we : id
```

```
wyn.zb = wyn.we
pisz  zb : id
}
```

Instrukcja dopoki umożliwia wielokrotne wykonywanie innych instrukcji. Ogólna postać tej instrukcji jest następująca:

dopoki <wyrazenie logiczne> : <instrukcja>

Podobnie jak w instrukcji gdy występująca po znaku : instrukcja może być zastąpiona blokem instrukcji w nawiasach { i }

Przykład:

```
$ss = 0
otworz  zb.asa = we
dopoki  $nr.we # 0 :
 {
 czytaj  we
 $ss = $ss + war.we
 }
```

6. Wiersz, kolumna.

Przyjęto następującą zasadę sporządzania raportów. Wszystkie wydruki są wstępnie przygotowywane w buforach o pojemności 1 wiersza drukarki (160 znaków). Bufory te nazywane w skrócie wierszami są ponumerowane 1,2.3... itd. Bufory mogą być podzielone na kolumny, których szerokość jest definiowana w programie. Kolumny są również ponumerowane 1,2,3...itd. Aby wprowadzić informację do bufora określa się zawsze numer wiersza oraz numer kolumny. Po przygotowaniu informacji do wydruku w buforach, zawartość buforów o podanych numerach jest wyprowadzana

na drukarkę lub ekran monitora.

Wiersze oraz kolumny są definiowane instrukcją **wzor**. Instrukcja ta służy również do wstępnego wypełniania bufora potrzebnymi znakami. Ogólna postać instrukcji **wzor** jest następująca:

wzor [<wiersz> / <kolumna> (<znak1> <znak2>

[, <szer kolumny>]...) | ,]...

W wyniku działania instrukcji zostanie utworzony bufor wiersza o numerze **<wiersz>** podzielony na kolumny o podanych numerach. Pierwszy znak (**<znak1>**) jest ogranicznikiem kolumny i służy do optycznego odzielenia poszczególnych kolumn. Do rysowania kresek pionowych oddzielających poszczególne kolumny używa się najczęściej znaków ! I | lub spacji. Drugim znakiem wypełniane jest wnętrze kolumny oraz dwa znaki po obu stronach kolumny. Najczęściej znakiem tym jest spacja. Do rysowania kresek poziomych w nagłówkach używa się - * lub = .Na kolumnę zostanie zarezerwowana ilość znaków określona parametrem **<szer.kolumny>** .

Jedną instrukcją wzór można definiować kilka buforów o kolejnych numerach wiersza. W tym przypadku zamiast jednego parametru **<wiersz>** podaje się dwa parametry oznaczające numer pierwszego i ostatniego wiersza oddzielone kropkami. W podobny sposób można określać kolumny o identycznym wzorcu.

Parametr **<szer.kolumny>** ma następującą postać:

[[<liczba> | <dana>[.<nzb>] | @<dana>[.<nzb>] |

"<tekst>"] | ,]...

Każda z 4 opcji definiuje w inny sposób szerokość kolumny. W

jednej definicji można użyć więcej niż jedną opcję. Również opcje tego samego typu mogą występować wielokrotnie. Przy takiej definicji zostanie przyjęta największa szerokość kolumny.

Pierwsza opcja definiuje bezpośrednio szerokość kolumny w znakach. Druga opcja służy do zarezerwowania miejsca do wydruku danej ze zbioru. Szerokość kolumny przyjmowana jest na podstawie parametrów zbioru określających format tej danej. Dwie ostatnie opcje służą do wypełniania nagłówka odpowiednim tekstem (28-mio znakową nazwą danej lub podanym bezpośrednio tekstem). Powodują one wpisanie tekstu do buforów wierszy o numerach (początkowym i końcowym), przy czym tekst zostanie tak rozmieszczony w poszczególnych wierszach (bez dzielenia słów) aby kolumna była możliwie wąska. W podobny sposób jest rozmieszczany tekst nazw danych w nagłówkach programu DRUK.

Szerokość kolumny może być zdefiniowana w programie tylko raz. W następnych instrukcjach wzor można podawać dla danej kolumny tylko znaki początkowe i końcowe.

Przykłady:

```
wzor  1..4 / 1 (! "Nazwisko",naz), 2..4 / 2 (! "Staz pracy",stp)
wzor 1 / 1..2 (--)
```

7. Wypełnianie buforów wiersza.

Do wypełniania buforów wiersza służy instrukcja:

```
wpisz  [ [ <wiersz> / <kolumna> ['<format>] = <wyr.arytm.> ¦
  <wiersz> / <kolumna> ['<pocz.>..<kon.>] : <wyr.tekstowe> ¦
  <wiersz> / <kolumna> ['<opcja>] : <wyr.tekstowe> ] ¦ , ]...
```

Instrukcja umożliwia wpisanie wartości wyrażenia

arytmetycznego, zgodnie z podanym formatem, do bufora wiersza o numerze <wiersz> i kolumny o numerze <kolumna> . Format określa się podając liczbę znaków przed kropką dziesiętną oraz po znaku . liczbę znaków po kropce. Dla zmiennych całkowitych nie podaje się ostatniego parametru i poprzedzającego go znaku. Napisanie bezpośrednio po formacie znaku # powoduje drukowanie wartości ze spacją co 3 znaki licząc od kropki (np. '5.3# daje wyrduk w postaci 99 999.999).

Instrukcja wpisz umożliwia ponadto wpisanie tekstu wyrażenia tekstowego. Opcjonalnie można określić początkowe i końcowe numery znaków w buforze, do którego jest kopiowany tekst, lub tylko numer znaku początkowego. Bezpośrednio po znaku ' można podać jeden ze znaków określających opcje:

> - przesunięcie wpisywanego tekstu maksymalnie w prawo,

* - umieszczenie tekstu na środku,

Zasady pisania wyrażeń arytmetycznych są identyczne jak we wcześniej omawianych instrukcjach podstawienia.

Przykład:

wpisz 1 / 1 '* : "Wynagrodzenia pracownikow"

wpisz 5 / 2 : nazw, 5 / 3 = wynagr, 5 / 4 '5.3 = wynagr + prem

8. Drukowanie raportów.

Drukowanie raportu polega na wyprowadzaniu zawartości buforów wierszy o podanych numerach. Służy do tego instrukcja **drukuj** :

drukuj <wiersz> [,<wiersz>]...

Wszystkie wiersze podane w jednej instrukcji są drukowane na

tej samej stronie.

Instrukcja **naglowek** w postaci:

naglowek <liczba wierszy na stronie> : <wiersz> [,<wiersz>]...

[/ <wiersz> [,<wiersz>]...]

[: <instrukcja>]...

określa numery wierszy, które należy wyprowadzić na początku następnej strony w momencie gdy wartość parametru **$nw** - numer wiersza przekroczy podaną wartość <liczba wierszy na stronie> . Opcjonalnie można podać (po znaku /) numery wierszy, które należy wydrukować na końcu każdej strony (tzw. stopka).

Parametr **$nw** jest automatycznie powiększany przez instrukcję **drukuj** . Wartość parametru **$nw** może być zmieniana w programie gdy zachodzi taka potrzeba. Wstępnie ustawiana duża wartość parametru **$nw** powoduje wydrukowanie nagłówka na początku pierwszej strony wydruku po wywołniu pierwszej instrukcji **drukuj** . Dlatego jeżeli trzeba umieścić jakiś tekst przed nagłówkiem należy zmienić wartość tego parametru.

W instrukcji **naglowek** można również określić po znaku : instrukcję lub blok instrukcji, które należy wykonać przed wyprowadzeniem kolejnego nagłówka na nowej stronie. W większości przypadków będzie to instrukcja wpisania do bufora jednego z wierszy nagłówka warości parametru **$ns** - numer strony.

9. Parametryzacja wydruków.

Programy pisane w języku generatora raportów **RAP** mogą być parametryzowane. Instrukcja **parametr** w postaci:

parametr <numer param.> : <nazwa param.> : <wartosc param.>

powoduje wyświetlenie na ekranie nazwy i po znaku : wartości parametru. Użytkownik może zachować wyświetloną wartość parametru wprowadzając pusty rekord lub wprowadzić nową wartość parametru. We wszystkich miejscach programu, w których wystąpi tekst:

%<nr parametru>

zostanie on zastąpiony tekstem <wartosc parametru>.

Przykład:

```
parametr 1 :numer kwartalu : 1
...
czytaj zb
gdy nrkw = %1
 {
 ...
 }
```

mgr inz. Jan Wegrzyn
Instytut Elektrotechniki Teoretycznej
i Miernictwa Elektrycznego
Politechnika Warszawska

SYSTEM OBSLUGI KATALOGU I PROWADZENIA DOKUMENTACJI BIBLIOTEKI

1. Wstep

System obslugi katalogu i prowadzenia dokumentacji biblioteki SKB-1 powstal w wyniku dostosowania do potrzeb bibliotecznych uniwersalnego systemu informacyjnego SIM-400R w Instytucie Elektrotechniki Teoretycznej i Miernictwa Elektrycznego Politechniki Warszawskiej.

Instytut ETiME posiada dwa minikomputery MERA-400 w dosc rozbudowanej konfiguracji (1 MB pomieci operacyjnej, 14 monitorow ekranowych oraz 2 stacje dyskow Winchester po 20 MB). Sluza one glownie do prowadzenia zajec dydaktycznych i prac naukowych. Duza moc obliczeniowa zainstalowanego sprzetu pozwala wykorzystac go takze do innych celow, m.in. postanowiono skomputeryzowac przy pomocy MERY-400 dzialalnosc biblioteki instytutowej.

W celu zmniejszenia kosztow oraz skrocenia czasu realizacji projektu postanowiono zakupic gotowe oprogramowanie: system informacyjno-wyszukiwawczy SIM-400R opracowany w Instytucie Informacji Naukowej, Technicznej i Ekonomicznej w Warszawie. Po zainstalowaniu ~~sprzetu~~ systemu okazalo sie jednak, ze wiekszosc uzytkownikow nie byla z niego zadowolona. Glowne zastrzezenia dotyczyly skomplikowanego sposobu posluiwania sie systemem, wymagania znajomosci systemu operacyjnego i oprogramowania systemowego MERY-400. Uzytkownik bez specjalnego przeszkolenia nie byl w stanie skorzystac nawet z najprostszych funkcji systemu.

Instytut ETiME nie mial mozliwosci przeszkolenia wszystkich uzytkownikow biblioteki, gdyz wiekszosc z nich to studenci pierwszych lat studiow. Konieczne bylo wiec przerobienie systemu SIM-400R w kierunku maksymalnego uproszczenia

kontaktu "czlowiek-maszyna". Wobec negatywnego stanowiska producenta w tej sprawie, zadanie to zostalo przeprowadzone silami wlasnymi instytutu.

2. System SIM-400R i jego zastosowanie w bibliotece

System informacyjno-wyszukiwawczy SIM-400R [1] jest pakietem programow przeznaczonych do tworzenia i przetwarzania baz danych. SIM-400R jest systemem uniwersalnym, narzucajacym tylko niewielkie ograniczenia na format przetwarzanych dokumentow. System realizuje nastepujace funkcje:

-zakladanie i aktualizacja baz danych;

-wyszukiwanie dokumentow w bazie wedlug zadanych kryteriow i ich wydruk;

-sortowanie i scalanie baz danych;

-realizacja obliczen z wykorzystaniem pol liczbowych w dokumentach

-generowanie slownikow i indeksow.

System SIM-400R jest przeznaczony glownie do pracy wsadowej. Dzialaniem poszczegolnych programow steruje ciag dyrektyw, ktore powinny byc wczesniej przygotowane przy pomocy edytora tekstu.

Podstawowe funkcje, ktore powinny byc realizowane przez system obslugi biblioteki to tworzenie i aktualizacja katalogu ksiazek, rejestru czytelnikow i rejestru wypozyczen oraz ich przegladanie wedlug roznych kryteriow. Przy pomocy systemu SIM-400R wszystkie te funkcje mozna wykonac, lecz posługiwanie sie nim jest niewygodne i uciazliwe. Potencjalny uzytkownik biblioteki, by sprawdzic czy ksiazka, ktorej szuka jest w bibliotece, musi najpierw przeczytac 8 stron instrukcji (tylko dla programu przeszukiwania). Dyrektywa okreslajaca kryterium przeszukiwania jest bardzo rozbudowana i umozliwia sformulowanie nawet dosc skomplikowanych warunkow, ale jednoczesnie forma najprostrzych nawet warunkow jest zlozona. Na przyklad, sprawdzenie czy w katalogu ksiazek jest pozycja "Matusiak Ryszard : Teoria pola" wymaga przygotowania nastepujacych dyrektyw:

/NAZWA/KATALOG-KSIAZEK

/SZUKAJ/MATUSIAK(AUT)&TEORIA(TYT)&POLA(TYT)

W systemie SIM-400R wszystkie przeszukiwania bazy danych odbywaja sie w

sposob sekwencyjny. W efekcie nawet dla niewielkiej biblioteki (ok. 5 tys. ksiazek) realizacja najprostrzego przeszukania katalogu trwa ponad minute. Reczne sprawdzenie w tradycyjnym katalogu trwa krocej. Oczywiscie, SIM-400R moze w ciagu minuty zrealizowac takie przeszukiwanie, ktore wykonane tradycyjnie trwaloby godzine, lecz te najprostrze operacje w przypadku biblioteki spotykane sa najczesciej.

Na podstawie doswiadczen z eksploatacji systemu SIM-400R sformulowano nastepujace wymagania, ktore musi spelniac system obslugi biblioteki:

- system musi byc dostepny dla uzytkownika bez specjalnego przeszkolenia; uzytkownik nie musi nic wiedziec o komputerach i systemach informacyjnych; wymagana jest tylko umiejetnosc posługiwania sie klawiatura;
- komunikacja uzytkownika z systemem powinna byc maksymalnie uproszczona (przynajmniej dla najczesciej wykonywanych funkcji);
- system powinien byc samodokumentujacy sie, tzn. powinien sam informowac uzytkownika, co i w jakiej formie nalezy wprowadzic;
- czas realizacji najprostrzych funkcji nie powinien przekraczac kilku sekund.

Wymagania te zostaly przyjete jako zalozenia projektowe przy tworzeniu systemu SKB-1.

3. Charakterystyka systemu SKB-1

System SKB-1 jest konwersacyjnym systemem obslugi katalogu i prowadzenia dokumentacji biblioteki. Jest systemem jednodostepnym, tzn. jednoczesnie moze byc wykonywana tylko jedna operacja na bazach danych. SKB-1 tworzy i obsługuje trzy bazy danych: katalog ksiazek, rejestr czytelnikow i rejestr wypozyczen. Rejestr ksiazek zawiera wszystkie informacje, ktore sa podane w karcie katalogowej ksiazki w tradycyjnym katalogu (autor, tytul, rok wydania, sygnatura). Rejestr czytelnikow zawiera informacje o czytelnikach biblioteki (imie i nazwisko, numer, miejsce pracy, telefon). W rejestrze wypozyczen zawarte sa dane o ksiazkach aktualnie wypozyczonych (sygnatura, numer wypozyczajacego, data zwrotu).

System SKB-1 realizuje nastepujace funkcje:

-tworzenie i aktualizacja katalogu ksiazek

-tworzenie i aktualizacja rejestru czytelnikow

-tworzenie i aktualizacja rejestru wypozyczen

-przeszukiwanie katalogu ksiazek - sprawdzanie czy dana ksiazka jest w bibliotece i jaki ma numer

-przeszukiwanie rejetru czytelnikow i wypozyczen:

-wyszukiwanie ksiazek wypozyczonych przez danego czytelnika

-wyszukiwanie czytelnika, ktory dana ksiazke wypozyczyl

-wyszukiwanie ksiazek, dla ktorych minela juz data zwrotu.

Dostep do niektorych funkcji (np. aktualizacja katalogu) jest zastrzezony dla personelu biblioteki i chroniony haslem.

Korzystanie z poszczegolnych funkcji odbywa sie poprzez odpowiedzi na zadawane przez system pytania. Jesli pierwszym wprowadzonym w odpowiedzi znakiem jest '?', system podaje co i w jakiej postaci nalezy teraz wprowadzic. Forma odpowiedzi jest maksymalnie uproszczona; zrezygnowano np. ze sformalizowanego jezyka opisu warunku przy przeszukiwaniu. Ogranicza to mozliwosci systemu; nie mozna np. znalezc w jednym kroku wszystkich ksiazek, w ktorych tytule wystepuje slowo "elektryczny" lub wydanych w roku 1987, trzeba zrobic dwa przeszukiwania.

Jezeli informacja wyprodukowana przez system w trakcie wykonywania zadanej funkcji przekracza 20 wierszy tekstu, dzielona jest na "strony" o takiej dlugosci (podstawowym urzadzeniem komunikacji z SKB-1 jest monitor ekranowy). Uzytkownik moze nastepnie przegladac strony w dowolnej kolejnosci. Moze takze skierowac cala informacje na drukarke.

4. Implementacja systemu SKB-1

W celu skrocenia czasu potrzebnego na stworzenie systemu SKB-1 starano sie w maksymalny sposob wykorzystac elementy systemu SIM-400r. Poniewaz producent nie zgodzil sie udostepnic wersji zrodlowych, mozna wiec bylo korzystac tylko z calych poszczegolnych programow pakietu SIM-400R.

Poszczegolne funkcje SKB-1 implementowano na dwa sposoby: wykorzystujac program SIM-400R, ktory realizuje te sama funkcje lub piszac je od nowa. W

pierwszym przypadku dopisano dwa moduly komunikacji z uzytkownikiem. Pierwszy modul zadaje pytania i zbiera na nie odpowiedzi. Nastepnie zapisuje na dysku dyrektywy sterujace dla odpowiedniego programu SIM-400R i wywoluje go. Drugi modul przeksztalca wyprodukowana informacje do postaci wygodniejszej dla uzytkownika. Tak zrealizowano m.in. program tworzenia i aktualizacji katalogu ksiazek.

W przypadku programu przeszukiwania katalogu ksiazek, takie rozwiazanie nie gwarantowalo odpowiedniej szybkosci. W systemie SIM-400R wszystkie przeszukiwania realizowane sa poprzez sekwencyjne przegladanie calej bazy danych. Wymaga to N porownan (N - liczba dokumentow w bazie). W systemie SKB-1 zastosowano przeszukiwanie metoda bisekcji, ktora wymaga co najwyzej log.N porownan. Metoda bisekcji wymaga jednak, by baza byla uporzadkowana (np. alfabetycznie). W przypadku katalogu ksiazek nie mozna tego zrealizowac, gdyz wyszukiwanie odbywa sie na podstawie wystepowania w poszczegolnych polach dokumentu podanych przez uzytkownika slow. Poniewaz niektore pola (np. pole tytulu) moga zawierac kilka slow, wymagaloby to utworzenia kilku (lub nawet kilkunastu) kopii katalogu odpowiednio uporzadkowanych. Wybrano wiec rozwiazanie z wykorzystaniem plikow indeksow. Plik indeksow zawiera adresy dyskowe uporzadkowanych alfabetycznie slow wystepujacych w danym polu. Ze wzgledu na oszczednosc miejsca na dysku, uzywane sa tylko trzy pliki indeksow, dla pola autora, tytulu i sygnatury. Jezeli uzytkownik poszukujac ksiazki nie poda danych dla tych pol, przeszukiwanie odbywa sie w sposob sekwencyjny.

Zastosowanie plikow indeksow powoduje, ze po kazdorazowej aktualizacji bazy danych trzeba je tworzyc od nowa. W przypadku biblioteki nie jest to krytyczne, gdyz aktualizacja odbywa sie dosc rzadko.

5. Uwagi koncowe

W chwili obecnej zrealizowano i zainstalowano system obslugi katalogu ksiazek (tworzenie, aktualizacja i przeszukiwanie). Aktualnie trwaja prace nad utworzeniem baz danych zawierajacych rejestr czytelnikow i rejestr wypozyczen.

Katalog ksiazek zawiera ok. 5 tys. pozycji, rejestr czytelnikow ok. 100, a rejestr wypozyczen ok. 700. Bazy danych dla tej ilosci informacji zajmuja ok. -

150 sciezek (800 kB) pamieci dyskowej. System SKB-1 potrzebuje jeszcze 50 sciezek (300kB) na pliki indeksowe oraz 100 sciezek (600 kB) na pliki robocze. W sumie potrzeba ok. 300 sciezek (1.8 mB) pamieci dyskowej.

System SKB-1 dziala pod kontrola systemu operacyjnego FLS.

Czas potrzebny na przeszukanie katalogu ksiazek wynosi kilka sekund, jezeli podano autora lub fragment tytulu oraz kilkadziesiat sekund w pozostalych przypadkach.

Aktualna wersja systemu SKB-1 nie jest wersja koncowa, ewentualne modyfikacje i rozbudowa nastapi po zebraniu opinii i uwag uzytkownikow.

6. Literatura

1. Bankowski J., Faber R., Fialkowski K., Furman M., Karpinski K., Maruszkiewicz M., Nowicki Z., Solak J: "Minikomputerowy system informacyjno-wyszukiwawczy SIM-400R"; Prace IINTE 36, Warszawa 1981

Piotr Nowak
Zakład Fototechniki
Politechnika Wrocławska

ZESTAW PROGRAMÓW DO OBLICZANIA FUNKCJI AUTOKORELACJI I WIDMA SPEKTRALNEJ GĘSTOŚCI MOCY. PRZYKŁADY ZASTOSOWAŃ.

Teoretyczne podstawy aplikacji metod numerycznej analizy szeregów czasowych do oceny własności strukturalnych obrazów fotograficznych, zostały sformułowane w latach pięćdziesiątych naszego wieku. Ówczesne środki pomiarowe i obliczeniowe silnie ograniczały szerokie stosowanie nowych i nie do końca poznanych metod. Istotnym przełomem w tej dziedzinie było opublikowanie w 1965 roku algorytmów szybkiego przekształcenia Fouriera /FFT/, pozwalających znacznie skrócić czas obliczeń. Opracowane wcześniej teorie zastosowań stanęły w świetle nowych sprawniejszych środków teoretycznych i technicznych. W tym czasie zaobserwowano gwałtowny rozwój zastosowań głównie w telekomunikacji i akustyce. Aplikacja nowych metod do oceny własności strukturalnych obrazów fotograficznych były czynione nieśmiało i z dużą dozą ostrożności. Obecnie, mimo dużej popularności metod cyfrowej analizy szeregów czasowych, dziedzina ta kryje w sobie wiele niewyjaśnionych problemów.

W 1981 roku w Zakładzie Fototechniki Politechniki Wrocławskiej również zostały podjęte prace nad zastosowaniem metod cyfrowej analizy szeregów czasowych do opisu własności strukturalnych obrazów fotograficznych. W związku z ograniczoną ilością informacji dotyczącą zastosowań konkretnych algorytmów, zdecydowano się na utworzenie zestawu programów umożliwiających obliczanie funkcji autokorelacji i widma mocy wszystkimi znanymi metodami. W ten sposób powstały dwa pakiety podprogramów napisanych w języku Fortran zawierających odpowiednio programy do obliczania funkcji autokorelacji /FAK/ i widma Wienera /WW/.

Pakiet FAK zawiera dwa programy:
pierwszy o nazwie AKD umożliwia obliczanie funkcji autokorelacji według algorytmu zapisanego w jej definicji /1/,

$$R_{xr} = 1/(N-r) \sum_{r=1}^{N-r} x_i * x_{i+r} \qquad (1)$$

$$r = 0,1,2,\ldots\ldots,m$$

gdzie: R_{xr} - estymator funkcji autokorelacji
N - ilość elementów zbioru danych X
m - zakres korelacji
r - wskaźnik aktualnego przesunięcia

drugi program o nazwie AKF umożliwia obliczanie funkcji autokorelacji metodą "rzeczywistego" przekształcenia Fouriera widma modułów /2/

$$R_{xr} = N/(N-r)\, F^{-1}[S_x] \qquad (2)$$

$$r = 0,1,2,\ldots\ldots,2N-1$$

gdzie: F^{-1} - operacja transformacji Fouriera
S_x - niewygładzone widmo mocy

Oba programy pozwalają opcjonalnie obliczać autokorelację lub autokowariancję w postaci funkcji liniowych lub cyklicznych.

Pakiet WW zawiera trzy podprogramy WWD, WWF i PER. Program WWD służy do obliczania widma spektralnej gęstości mocy poprzez tzw. cosinusowe przekształcenie Fouriera funkcji autokorelacji obliczonej metodą "z definicji" /3/,

$$S_x = 4 \int_0^{\infty} R_{xr} \cos 2\pi v r \, dr \qquad (3)$$

gdzie: v - częstotliwość przestrzenna.

Program WWF umożliwia obliczanie widma mocy jako część rzeczywistą przekształcenia Fouriera funkcji autokorelacji otrzymanej drogą FFT /4/,

$$S_x = F[uR_{xr}] \qquad (4)$$

gdzie: u - funkcja wagowa zapobiegająca tzw. "przeciekowi" widma /okno filtracyjne w dziedzinie czasu/.

Program PER pozwala obliczać widmo Wienera w postaci periodogramu stanowiącego niewygładzony estymator widma modułów /5/,

$$S_x = (2\Delta t/N)\,|X_x|^2 \qquad (5)$$

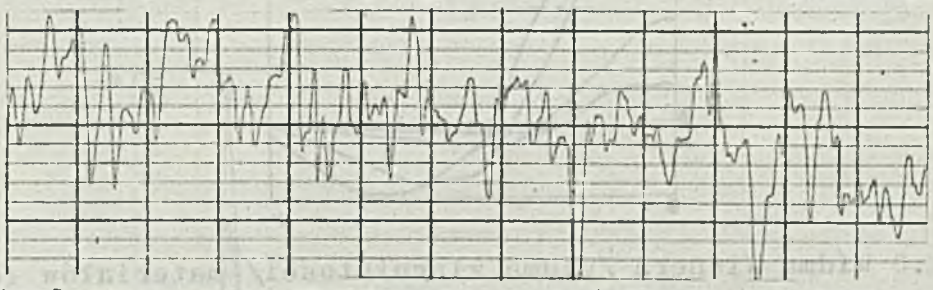
gdzie: dt - interwał pomiarów

X_x - widmo modułów.

W pracach zastosowano wyjątkowo sprawny algorytm szybkiego przekształcenia Fouriera opublikowany w 1976 roku przez D.M. Monro w miesięczniku Applied Statistics.

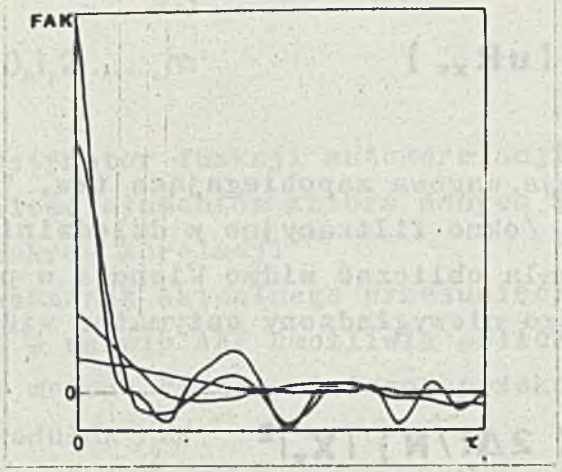
Wszystkie programy zostały uruchomione i przetestowane w systemie operacyjnym SOM3 a następnie przeniesione pod system CROOK gdzie funkcjonują już od kilku lat.

Przykłady działania programów zilustrowano poniżej.



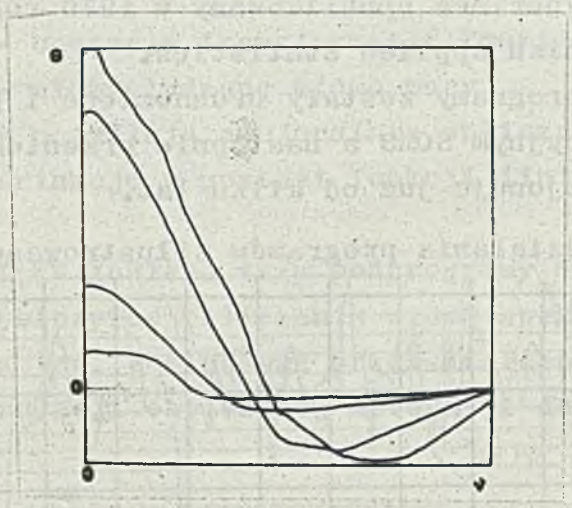
Rys.1 Analogowy zapis fluktuacji gęstości optycznej obrazu fotograficznego.

Na rysunku 2 przedstawiono znormalizowane /względem siebie/ funkcje autokorelacji, fluktuacji gęstości optycznej struktury ziarnistej, czterech różnych materiałów fotograficznych, otrzymane przy pomocy programu AKD.



Rys.2 Funkcje autokorelacji czterech różnych materiałów fotograficznych.

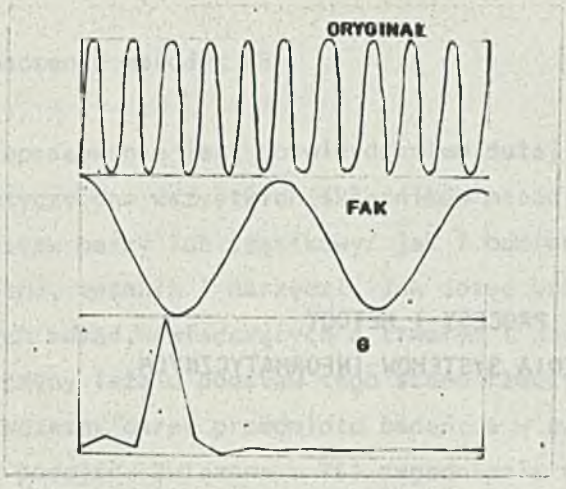
Rysunek 3 ilustruje widma mocy tych samych materiałów fotograficznych. Estymatory obliczono przy pomocy programu WWD.



Rys.3 Widma Wienera /widma ziarnistości/ materiałów fotograficznych różniących się strukturą ziarnistą.

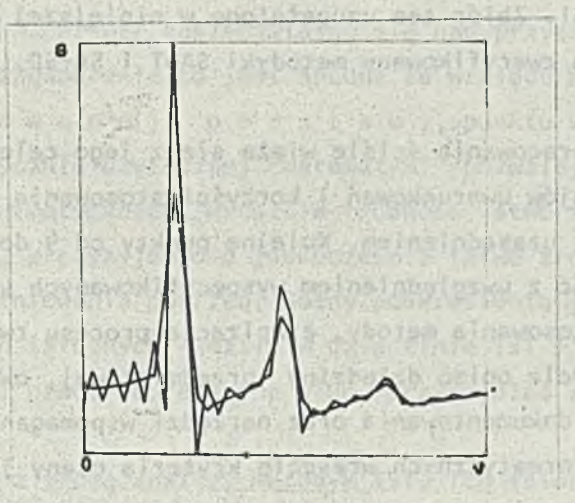
Z bezwzględnych wartości oraz kształtu funkcji FAK i WW wnioskować można o przebiegu procesów fizykochemicznych jakie zachodzą w czasie produkcji, obróbki i starzenia się materiałów fotograficznych.

Górna część rysunku 4 przedstawia mikrodensytometryczny zapis zmian gęstości optycznej rastra graficznego o nieznanej liniaturze. Obliczając widmo mocy takiego sygnału /dolna część rysunku/ można w prosty sposób określić, z położenia maksimum funkcji, liniaturę rastra.



Rys.4 Rozkład gęstości optycznej, funkcja autokorelacji i widmo mocy rastra graficznego.

Rysunek 5 przedstawia widmo Wienera mikrofotometrycznego zapisu fluktuacji gęstości optycznej jaki powstał w czasie obserwacji skali mikroskopowej. Z analizy funkcji wnioskujemy że, w obserwowanym obrazie występują różnej szerokości linie /różna szerokość połówkowa pasm/ pojawiające się z częstotliwością przestrzenną wynikającą z położenia pasma oraz intensywnością proporcjonalną do jego wysokości.



Rys.5 Wygładzone i niewygładzone widmo mocy skali mikroskopowej.

Stanisław Wrycza
Uniwersytet Gdański
81-824 Sopot
Armii Czerwonej 101

PROCESY I METODY TWORZENIA SYSTEMOW INFORMATYCZNYCH

1. Wprowadzenie

Celem niniejszego opracowania jest dokonanie przeglądu aktualnego stanu w i e d z y w sferze tworzenia systemów informatycznych TSI poprzez przeprowadzenie a n a l i z y p o r ó w n a w c z e j współczesnych metodyk TSI. Bazę badawczą stanowiły liczne, związane z tymi podejściami pozycje bibliograficzne głównie literatury zagranicznej oraz wcześniejsze badania własne autora. Przedmiotem badań jest przede wszystkim 13 m e t o d y k objętych badaniami CRIS /Comparative Review of Information Systems Design Methodologies/ IFIP TC 8. Dobór metodyk do porównania CRIS był poprzedzony wszechstronną, kilkuetapową analizą, dokonaną przez kompetentny zespół ekspertów po uwzględnieniu ponad pięćdziesięciu podejść. Wybrane metodyki są reprezentatywne dla różnych kierunków w ramach TSI. Zbiór ten uzupełniono w niniejszej pracy o bardzo pragmatyczne, wielokrotnie zweryfikowane metodyki SADT i SA-SD.

U k ł a d i treść opracowania ściśle wiążą się z jego celem. Po obszernym omówieniu celów, rodzajów uwarunkowań i korzyści stosowania metodyk TSI dokonano ich doboru wraz z uzasadnieniem. Kolejne punkty od 4 do 9 to analiza porównawcza podejść z uwzględnieniem wyspecyfikowanych uprzednio kryteriów jak: cele i zastosowania metody, etapizacja procesu tworzenia systemów informatycznych, modele opisu dziedziny przedmiotowej, określenie podstawowych metod i technik dokumentowania oraz narzędzi wspomaganego komputerem tworzenia systemów informatycznych wreszcie kryteria oceny jakości systemu. Ostatnie dwa punkty opracowania są syntezą dokonanych wcześniej porównań oraz wnioskami i zaleceniami dla projektantów systemów. Bibliografia obejmuje wyszczególnienie opracowań głównie porównawczych, które zawierają obszerne zestawy literatury źródłowej.

2. Rodzaje i znaczenie metodyk TSI

Niniejsze opracowanie jest potwierdzeniem dużej różnorodności w zakresie metodyk TSI. Dotyczy ona wszystkich składników metodyki i to w odniesieniu do ich zakresu /zestaw pełny lub cząstkowy/ jak i odmienności poszczególnych formalizmów, metod, technik i narzędzi. Jak dotąd brak jest uogólnionych powszechnie uznanych zasad, świadczących o trwałym i naukowym charakterze dziedziny TSI. Jakie przyczyny leżą u podstaw tego stanu rzeczy? Wydaje się, że jest to n o w o ś ć i wczesny okres przedmiotu badań, a w związku z tym pewna żywiołowość propozycji podejść. Związane z TSI zagadnienia należą do problemów n i e u s t r u k t u r y z o w a n y c h a więc złożonych zarówno w zakresie ich definiowania w konsekwencji rozwiązywania. Inne kwestie mające wpływ na aktualny stan rzeczy to: specyfika dziedzin przedmiotowych poddawanych procesowi modelowania, szybkie zmiany w dziedzinie inżynierii oprogramowania i jej wpływ na konstruowanie zautomatyzowanych narzędzi wspomagania procesu TSI jak również dążenie do nie zawsze uzasadnionej oryginalności poszczególnych autorów, konstruujących /n + 1/-wszą metodykę.

Niewątpliwie obiektywną przesłanką wspomnianej różnorodności są negatywne d o ś w i a d c z e n i a we wdrażaniu kolejnych podejść i stąd naturalne próby ich u d o s k o n a l e n i a bądź oparcia na innych podstawach teoretycznych. Ceri /CERI-86/ zastanawiając się nad przyczynami omawianej sytuacji, stwierdza, iż zagadnienie to jest trudne ze względu na brak d o b r z e z d e f i n i o w a n e j p o d s t a w ,, punktu wyjścia cyklu życia systemu, brak praktycznie użytecznej "matematyki" pozwalającej na wyrażanie p o t r z e b informatycznych, wreszcie trudność stworzenia zaawansowanego, poprawnego logicznie, precyzyjnego a jednocześnie łatwo zrozumiałego dla użytkownika formalizmu definiowania potrzeb. Godny podkreślenia jest sam fakt określenia nierozwiązanych istotnych kwestii w dziedzinie TSI - narastające z czasem doświadczenia i prace teoretyczne winny przyczyniać się do ich rozwiązania.

W związku z próbą analizy metodyk TSI, ich zbiorowość należy s k l a s y f i k o w a ć. Z podanych wyżej względów nie jest to zadanie łatwe. Zaproponowano następujące kryteria oceny: podejście do procesu TSI, definiowanie danych bądź procesów w projekcie, kierunek TSI, stopień aktywności procesu TSI.

Pierwsze podstawowe kryterium oznacza wyodrębnienie m e t o d y k t e c h n i c z n y c h i s p o ł e c z n y c h /CER 86 /. Nazwy są tu umowne, choć oddają one istotę podejść. Metodyki techniczne ukierunkowane są na realizację dobrze ustrukturyzowanego procesu TSI, z pełnymi i sformalizowanymi modelami opisu rzeczywistości. Podejście opiera się na założeniu, ża analityk ma n e u t r a l n y wpływ na organizację, która będzie współpracować z nim w procesie TSI. Z kolei metodyki społeczne akcentują organizacyjne i ludzkie problemy związane z TSI. Celem tego procesu jest zrozumienie roli systemu informacyjnego w ramach systemu społecznego i oddziaływanie na obydwa. Rola analityka jest tu bardziej a k t y w n a. Powodzenie strategii TSI jest w tym przypadku uzależnione od umiejętności specyfikacji kluczowych uwarunkowań organizacyjnych i możliwości oddziaływania na nie. Metodykom technicznym zarzuca się iż stanowią one swoiste "książki kucharskie", które mogą ominąć istotę problemu. Natomiast podejście społeczne nie pozwala na tak dokładne sprecyzowanie potrzeb informatycznych aby stanowiły one podstawę dla realizacji kolejnych faz cyklu życia systemu. Praca niniejsza dotyczy przede wszystkim metodyk pierwszej grupy, choć najbardziej pożądana byłaby integracja obydwu podejść.

Metodyki zorientowane na d a n e dotyczą strukturyzacji danych użytkowanych w organizacji natomiast zorientowane na p r o c e s y są określane przez przepływ informacji. Na tej podstawie, w oparciu o potrzeby użytkowników specyfikowane są dane elementarne. Omawiane metodyki stanowią zazwyczaj kompromis, o różnej proporcji, pomiędzy rozwiązaniami ukierunkowanymi na dane i procesy.

Podstawą kolejnego podziału są relacje pomiędzy systemem rzeczywistym /dziedziną przedmiotową/ a systemem informatycznym /LOCK-86/. Pierwszy rodzaj związku to "o r g a n i z a c y j n e o d w z o r o w a n i e" zakładające pasywną rolę systemu informatycznego. Decyzje i działania są podejmowane w systemie rzeczywistym. W związku z tym system informatyczny musi być prawdziwym o d z w i e r c i e d l e n i e m systemu rzeczywistego aby być efektywnym. Przeciwne podejście "o r g a n i z a c y j n e g o s t e r o w a n i a" zakłada wyróżnialny system ludzkiego sterowania w którym podejmuje się decyzje i działania o d d z i a ł u j ą c e na dziedzinę przedmiotową a obejmujący system informatyczny jako swą integralną część.
Informacja jest wybierana, opracowywana i grupowana zgodnie z tym celem. Nacisk kładzie się mniej na precyzyjny opis świata rzeczywistego a bardziej

na określanie p o t r z e b i n f o r m a t y c z n y c h. W polskiej literaturze określa się te dwa rodzaje podejść mianem aktywnego i pasywnego /NIED 77/.

Wreszcie ostatnie z wymienionych kryteriów oznacza wyodrębnienie metodyk z s t ę p u j ą c y c h /top-down/ i w s t ę p u j ą c y c h /bottom - up/. Podejście zstępujące oznacza tworzenie systemu poprzez stopniowe, hierarchiczne, analityczne wyodrębnianie jego składników do podstawowego poziomu szczegółowości. Podejście wstępujące z kolei polega na stopniowym opracowywaniu syntezy systemu poprzez integrację elementów systemu począwszy od poziomu podstawowego.

Analizując proponowane przez środowiska akademickie oraz firmy doradcze metodyki można sformułować wniosek, iż czysta klasyfikacja dotyczy pierwszego kryterium podziału na podejścia techniczne i społeczne. Omawiane niżej metodyki stanowią zazwyczaj swoisty k o m p r o m i s, o różnym udziale procentowym, pomiędzy specyfikacją danych i procesów, aktywnego i pasywnego oddziaływania na dziedzinę przedmiotową oraz wstępującego bądź zstępującego toku projektowania.

Dotychczasowe rozważania skłaniają ku próbie określenia w y m a g a ń odnośnie optymalnej metodyki, które winna ona spełnić. Wydaje się, iż najistotniejsze z nich to:

- metodyka winna objąć c a ł y c y k l życia systemu od analizy systemu do adaptacji i modyfikacji przy jednoczesnym umożliwieniu płynnych p r z e j ś ć pomiędzy poszczególnymi fazami,
- procedura TSI winna być wsparta r ó ż n o r o d n y m i, dostosowanymi do specyfiki podejścia, m e t o d a m i, technikami i narzędziami komputetowymi ułatwiającymi zrozumienie problemów, ich analizę i rozwiązanie,
- metodyka winna ułatwić p o r o z u m i e w a n i e się pomiędzy różnymi grupami zawodowymi tworzącymi nowy SI - dotyczy to zwłaszcza wstępnych faz procesu TSI gdzie winno być zaoferowane narzędzie - formalizm - wspólny język informatyków i użytkowników,
- metodyka winna być stosunkowo ł a t w a do opanowania i stosowana do d u ż e j klasy problemów oraz zawierać mechanizmy ewolucyjności i modyfikowalności.

Można wymienić cały szereg innych cech rzutujących na "optymalność" metodyki lecz mogą one zawężać pole twórczych poszukiwań w tej dziedzinie.

Badania są tu stymulowane przede wszystkim oczekiwanymi k o r z y ś c i a m i stosowania metodyki, do których należą: ułatwienie kierowania i tworzenia złożonych systemów, przyspieszenie tego procesu, konstruowanie łatwo eksploatowanych /przyjaznych user-friendly/ systemów.

Mimo powyższych uwarunkowań oraz intensywnych prac szeregu grup badawczych na świecie notuje się stosunkowo niewielką liczbę zastosowań, przy czym implementacje dotyczą grupy kilku zaledwie /por. punkty 3 i 4/ podejść. U podstaw tego stanu rzeczy leży kilka p o w o d ó w , z których najważniejsze to:

- duży zakres szybko dezaktualizującej się, trudnej do sterowania, papierowej dokumentacji wymaganej w procesie TSI,
- długi okres c z a s u pomiędzy zainicjowaniem a wdrożeniem projektu systemu,
- stosunkowo niewielka ilość komputerowo wspomaganych n a r z ę d z i TSI - narzędzia dostępne nie prezentują wysokiego poziomu jakości,
- obiektywne trudności i subiektywna niechęć do u c z e n i a s i ę i w konsekwencji s t o s o w a n i a metodyk ze względu na ich złożoność czy wysoki poziom zaawansowania,
- większość z metodyk to narzędzia nowe, znajdujące się w fazie rozwoju,
- r ó ż n o r o d n o ś ć rozwiązań utrudnia dobór właściwego podejścia do strukturyzacji konkretnej dziedziny przedmiotowej.

3. Dobór kryteriów ocen metodyk

Każda próba oceny określonej zbiorowości metodyk winna wychodzić w swej istocie z przesłanek obiektywnych. Wybrany, ograniczony z różnych względów zestaw kryteriów winien w najbardziej pełny sposób oddać specyfikę każdej rozpatrywanej metodyki, jej charakterystykę. Ponieważ wciąż brak jest ogólnych powszechnie uznanych teoretycznych podstaw tworzenia systemów informatycznych różnorodność proponowanych rozwiązań indywidualnych utrudnia przyjęcie ogólnie akceptowanego

wzorca. Spod zestawu przyjętych charakterystyk wymykać będą się pewne specyficzne cechy poszczególnych podejść toteż niezbędnym uzupełnieniem wszelkich analiz porównawczych w tej dziedzinie muszą być szczegółowe ich badania. W niniejszym zestawieniu porównawczym przyjęto następujące kryteria oceny metodyk TSI:

A. Cel, przeznaczenie, zastosowania, doświadczenia w użytkowaniu metodyki.
B. Etapizacja procesu tworzenia systemu.
C. Model - sposób reprezentacji rzeczywistości, istotne kategorie opisu statyki i dynamiki badanej dziedziny przedmiotowej.
D. Dokumentacja - podstawowe metody i techniki.
E. Narzędzia wspomaganego komputerem tworzenia systemu informatycznego w tym języku definiowania systemu.
F. Kryteria oceny jakości projektu systemu i procesu tworzenia.

Aktualnie w literaturze przedmiotu można zapoznać się z mniej lub bardziej precyzyjnymi opisami ponad 50 metodyk / CERI 86 /. Prezentują one różną wartość i poziom merytoryczny - trudno je wszystkie poddać analizie. Dla celów niniejszej pracy wybrano 15 podejść: 13 metodyk objętych charakterystyką CRIS jak również SADT oraz SA - SD, ze względu na ich użyteczność praktyczną i dobrą charakterystyką bibliograficzną. Tak więc przedmiotem analizy porównawczej będą następujące metodyki: ACM,PCM, CIAM, D2S2, DADES, EDM, IML, ISAC, ISSM, NIAM, REMORA, SADT, SA-SD, SDLA, SYSDOC, USE.

Kolejne punkty opracowania stanowią charakterystyki wymienionych metodyk w aspekcie podanych kryteriów oceny.

4. Cele i zastosowania metodyk

Mimo, iż w prezentowanych wyżej opisach często powtarza się słowo m e t o d y k a /czasem metodologia/ nie zawsze definiuje się czym ona jest, jakie są jej cele. Opis poszczególnych narzędzi, definiowanie aparatu terminologicznego przesłania bardziej generalne kwestie. Ogólną definicję metodyki TSI podaje Wasserman /WASS 83 /traktując ją jako z e s t a w m e t o d i z a u t o m a t y z o w a n y c h n a r z ę d z i dla TSI. Definicja ta jest uszczegóławiana poprzez określenie celu metodyki. Jest nim /REMORA/ stworzenie przez użytkowników, analityków i projektantów spójnych, zupełnych i integralnych

systemów informatycznych. Oznacza to rozwiązanie /ISSM/ dwu problemów związanych z TSI: kierowania projektem oraz użytkowania różnorodnych metod i narzędzi dla projektowania i wdrażania SI. Całość tych działań wpływa na podniesienie efektywności realizacji całego cyklu TSI.

Charakteryzowane metodyki powstały bądź w środowisku uniwersyteckim, a k a d e m i c k i m bądź p r o f e s j o n a l n y m informatycznych organizacji doradczych. Pierwsza grupa podejść to przede wszystkim metodyki ACM, PCM, CIAM, DADES, EDM, IML, ISSM, REMORA i SDLA natomiast druga grupa obejmuje D2S2, NIAM, SASD, SADT i SYSDOC. Metodyki USE i ISAC były tworzone na pograniczu obydwu środowisk. Podział ten jest znaczący również w sferze z a s t o s o w a ń - metodyki akademickie są raczej koncepcjami wprawdzie o wyraźnych podstawach naukowych lecz bez wdrożeń bądź z wdrożeniami eksperymentalnymi, podczas gdy z podejściami profesjonalnymi wiążą się liczne implementacje w organizacjach gospodarczych i innych instytucjach. Największa liczba zastosowań dotyczy metodyk: D2S2, ISAC, SA-SD, SADT i SYSDOC. Podejścia te były skonstruowane w pierwszej połowie lat siedemdziesiątych natomiast metodyki akademickie na przełomie lat sześćdziesiątych i osiemdziesiątych. K o r z y ś c i, które praktycy, użytkownicy metodyk uzyskują poprzez ich stosowanie są następujące /MADD 83/:

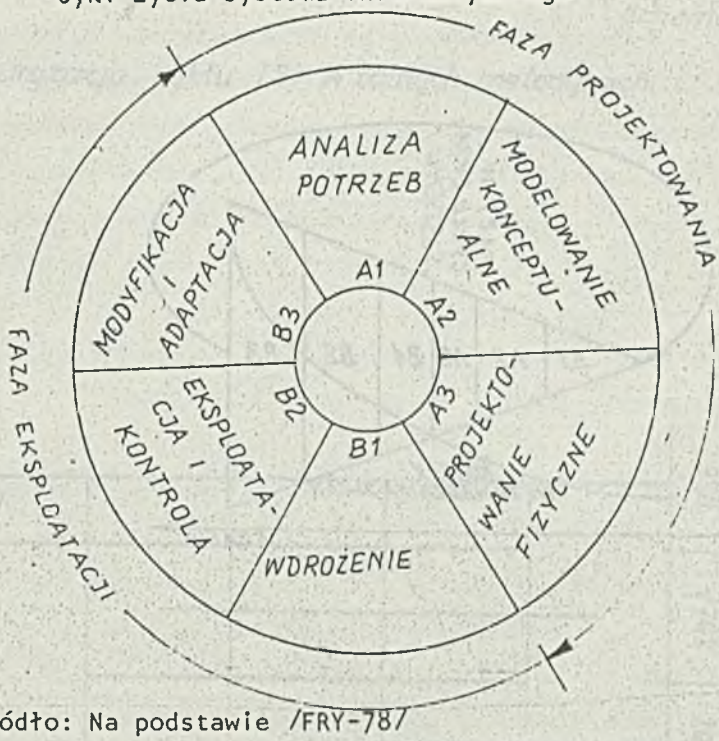
- precyzyjne uchwycenie potrzeb w zakresie przewidywanego zastosowania
- usystematyzowane prowadzenie procesu TSI w taki sposób, że postęp może być łatwo oceniony i kontrolowany w cyklu życia systemu
- tworzenie systemów, które realizują postawione przed nim cele
- minimalizacja zmian w fazie wdrożenia poprzez zanalizowanie i rozwiązanie wszystkich decyzji projektowhych na wcześniejszych etapach
- wykonanie cyklu TSI w realistycznych zakresach czasu, kosztów i zasobów
- opracowanie dobrze udokumentowanych, łatwo aktualizowanych systemów.

5. Etapizacja procesu tworzenia systemów informatycznych

Każda metodyka TSI winna wspomagać realizację wszystkich etapów tzw. c y k l u ż y c i a s y s t e m u, składającego się z fazy projektowania i użytkowania. Szczegółowo stadia cyklu z punktu widzenia projektanta i użytkownika systemu przedstawiają się jak na schemacie 1. Pomija się tu szerszą charak-

Schemat 1

Cykl życia systemu informatycznego



Zródło: Na podstawie /FRY-78/

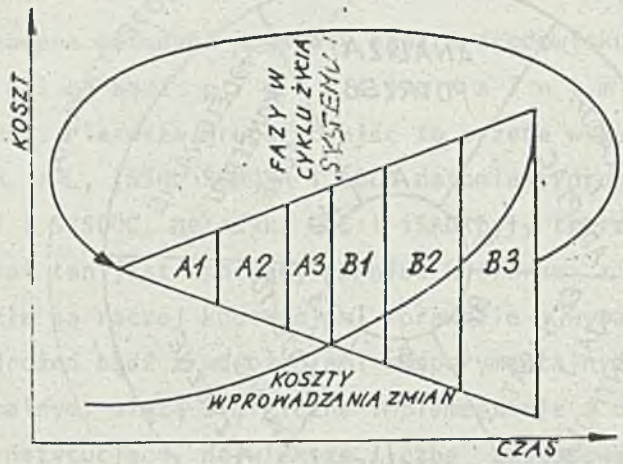
terystykę poszczególnych faz zawartą m.in. w pracy /WRYC 85 /.

Zarówno czas tworzenia systemu jak i jego jakość uzależnione są od poprawności i zupełności e t a p ó w w s t ę p n y c h związanych z analizą potrzeb i modelowaniem konceptualnym. Błędy popełnione w tych fazach są zwielokrotniane czasowo i kosztowo w stadium projektowania, fizycznego bądź w trakcie wdrażania /por. schemat 2/. Powyższa zależność jest jednym z podstawowych stymulatorów potrzeby rozwoju metodologii tworzenia systemów informatycznych.

W omawianych metodykach proces TSI jest u s t r u k t u r y z o w a n y w różnorodny sposób - zarówno jeśli chodzi o zakres jak i terminologię. Interesującą próbę ujednolicenia spojrzenia na wycinek procesu TSI podjęto w pracy / BRAN 83 /, w której cykl życia systemu podzielono na następujące f a z y:

Schemat 2

Cykl życia systemu a koszty wprowadzania zmian



Zródło: Na podstawie /ROSE 82/

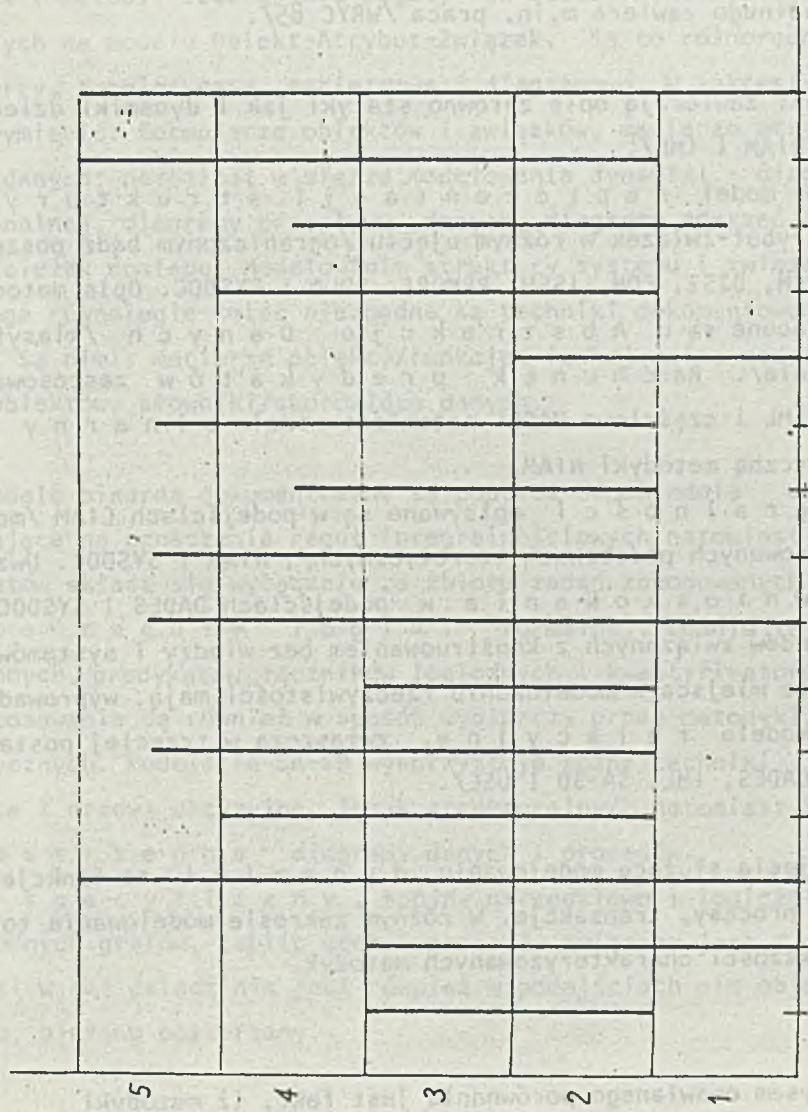
1. Studium procesów i celów
2. Modelowanie i specyfikacja systemu
3. Projekt logiczny systemu
4. Projekt szczegółowy
5. /Przygotowanie/ wdrożenia.

Analiza porównawcza poszczególnych metodyk w odniesieniu do powyższej struktury cyklu zaprezentowano na układzie współrzędnych na schemacie 3.

Pełny cykl życia systemu obejmują w zasadzie jedynie metodyki D2S2 oraz USE, zbliżony zakres uwzględniony jest w podejściach ISAC, SYSDOC, EDM, ISSM i REMORA. Pozostałe metodyki ujmują wybrany, często ograniczony w y c i n e k cyklu. Kilka podejść, a mianowicie CIAM, D2S2, ISAC, SADT, częściowo SDLA oraz USE zawierają narzędzia i metody analizy potrzeb informatycznych.

Schemat 3

Strukturyzacja cyklu TSI w różnych metodykach



Źródło: Na podstawie [BRAN-83]

6. Modele opisu dziedziny przedmiotowej

Zagadnienie to ściśle wiąże się z podstawami teoretycznymi tworzenia systemów informatycznych i to zarówno z zakresu modelowania statyki jak i dynamiki badanej dziedziny przedmiotowej. Dziedziną, która reprezentuje najbardziej zaawansowane rozwiązania w tym zakresie jest modelowanie koncepturalne. Opis statyki i dynamiki badanego wycinka rzeczywistości, noszący miano modelu bądź schematu konceptualnego może być dokonany dzięki następującym podejściom: Obiekt-Atrybut-Związek /czy Relacja/ /WRYC-87/a , Binarne, Rachunek Predykatów bądź Abstrakcje Danych.
Bardziej pogłębioną analizę poszczególnych podejść teoretycznych podstaw modelowania konceptualnego zawiera m.in. praca /WRYC 85/.

Omawiane metodyki zawierają opis zarówno statyki jak i dynamiki dziedziny przedmiotowej /poza NIAM i IML/.
Najbardziej popularny model reprezentacji struktury systemu to Obiekt-Atrybut-Związek w różnym ujęciu /ograniczonym bądź poszerzonym/ w metodykach - ACM/PCM, D2S2, EDM, ISSM, REMORA. SDLA i SYSDOC. Opis metodyk ACM/PCM i CIAM wzbogacone są o Abstrakcje Danych /klasyfikacja, agregacja i uogólnienie/. Rachunek predykatów zastosowano w podejściach CIAM, IML i częściowo DADES natomiast model binarny stanowi podstawę teoretyczną metodyki NIAM.
Więzy integralności opisywane są w podejściach CIAM /metodyka o najbardziej zaawansowanych podstawach teoretycznych/, NIAM i SYSDOC. Uwzględnienie reguł wnioskowania w podejściach DADES i SYSDOC zbliża je do aktualnych trendów związanych z konstruowaniem bez wiedzy i systemów ekspertowych. Znaczące miejsca w modelowaniu rzeczywistości mają, wyprowadzone z teorii baz danych modele relacyjne, zwłaszcza w trzeciej postaci normalnej /metodyki DADES, IML, SA-SD i USE/.

Konstrukcje, pojęcia służące modelowaniu dynamiki to funkcje, zdarzenia, operacje, procesy, transakcje. W różnym zakresie modelowanie to stosowane jest w większości charakteryzowanych metodyk.

Interesującym rysem omawianego porównania jest fakt, iż metodyki komercyjne o dużej liczbie wdrożeń jak ISAC, SADT, częściowo SA-SD i USE opierają się na własnych, autonomicznych konstrukcjach i modelach

posiadających walor użyteczności. Są one ukierunkowane na specyficzne metody i techniki dokumentowania analizy i projektowania systemów.

7. Podstawowe metody i techniki dokumentowania

Zestaw metod i technik dokumentowania systemu jest ogólnie różnorodny i bogaty choć dla poszczególnych metodyk zróżnicowany ilościowo i treściowo. Generalnie można je podzielić na ręczne /"papierowe"/ i zautomatyzowane. Druga grupa zostanie szerzej scharakteryzowana w punkcie 4.6. Zgodnie z podziałem modelowania rzeczywistości na modelowanie statyki i dynamiki opracowano odpowiednie techniki i metody. Szczególnie bogaty ich zestaw wykorzystuje się w metodykach bazujących na modelu Obiekt-Atrybut-Związek. Są to różnorodne techniki formularzy, tabelaryczne, macierzowe i diagramowe. W zakresie opisu struktury można wymienić: formularze obiektów i związków, macierze atrybutów, diagramy modeli danych; natomiast w sferze modelowania dynamiki - diagramy dekompozycji funkcjonalnej, diagramy przepływu danych, diagramy zdarzeń /sieci Petri/, diagramy ścieżek dostępu. Modelowanie struktury systemu i związanych z nim procesów przebiega równolegle toteż niezbędne są techniki dokumentowania wzajemnych powiązań. Są nimi: macierze obiekty/funkcje, funkcjonalne modele danych, cykle życia obiektów, słowniki/skorowidze danych.

Modele binarne dokumentowane są poprzez odpowiednie diagramy pozwalające na oznaczenie reguł integralnościowych natomiast opis w Rachunku Predykatów składa się wyłącznie ze zbioru zadań zakodowanych w języku formalnym opartym o reguły logiki formalnej. Zdania te złożone są z: nazw i zmiennych, predykatów, łączników logicznych i kwantyfikatorów. Wymienione techniki stosowania są również w sposób wybiorczy przez metodyki o innych podstawach teoretycznych. Podejście SA-SD wykorzystuje różne techniki decyzyjne /tablice i drzewa decyzyjne, język strukturalny/, natomiast SADT autonomicznie, przestrzenne diagramy danych i procesów.
Bardzo specyficzny, spójny narzędziowo i logicznie zestaw technik - różnorodnych grafów, tablic cech i procesów związany jest z metodyką ISAC. Wachlarz narzędzi w tej dziedzinie jest również w podejściach nie objętych niniejszym studium, bieżąco poszerzany.

8.Wspomagane komputerem TSI

Wzrost i potrzeby aktualizacji systemów informatycznych narzucają konieczność tworzenia komputerowych narzędzi TSI. Proces konstruowania oparty o techniki manualne, papierowe staje się w przypadku modelowania złożonych dziedzin przedmiotowych niesterowalny. Rozwój techniki mikrokomputerowej również skłania do poszukiwań bardziej nowoczesnych i efektywnych rozwiązań, do poszerzania środowiska tworzenia systemów informatycznych o zautomatyzowane narzędzia.

Wspomagane komputerowo narzędzie tworzenia systemów informatycznych stanowi spójny zespół programów umożliwiających:

- kompilację opisów składników systemu, dokonanych w odpowiednim języku opisu systemu
- interaktywny sprzęg między użytkownikiem a narzędziem w definiowaniu systemu i wyszukiwaniu jego składników
- edycję aktualnej dokumentacji systemu oraz związanych z nim zestawień analitycznych i diagnostycznych
- stałą kontrolę poprawności i integralności tworzonego opisu systemu w trakcie realizacji operacji tworzenia, skreślania, modyfikacji i wyszukiwania składników systemu.

Niniejsze studium nie obejmuje klasycznych narzędzi wspomagania procesu TSI typu pakiet PSL/PSA /TEIC 77/ czy INCOD - DTE /CERI 83/. W większości omawianych metodyk, jakkolwiek zagadnienie to jest sygnalizowane i doceniane, brak jest propozycji konkretnych rozwiązań w zakresie automatyzacji procesu TSI. Istniejące propozycje można podzielić na dwie grupy:

- języki opisu systemu, definiowania potrzeb czy schematu konceptualnego, umożliwiające również generowanie zestawień analitycznych w podejściach DADES, ISSM, NIAM, SDLA, REMORA, częściowo D2S2 /słownik/skorowidz danych/,
- zaawansowane narzędzia wspomaganego komputerem TSI obejmujące oprócz języka opisu systemu cały szereg wyżej wymienionych funkcji - należą tu metodyki

ACM/PCM, SADT /System SAS/, SYSDOC i USE oraz ISAC[1/].

Zagadnienia wspomagania komputerowego są istotnym trendem współczesnej teorii i praktyki TSI. Dokonują się tu szybkie zmiany i udoskonalenia wymagające bieżącej obserwacji i analizy.

9. Kryteria oceny jakości systemu

Określenie kryteriów oceny tworzonych systemów bądź całego procesu TSI jest przedmiotem zainteresowania w sposób bezpośredni lub pośredni, każdej z omawianych metodyk. Najczęściej powtarzają się dwa kryteria: s p ó j n o ś c i i k o m p l e t n o ś c i /zupełność/. Pierwsza oznacza wzajemną niesprzeczność wszystkich składników systemu. Z kolei realizacja kryterium kompletności wymusza precyzyjne zdefiniowanie wszystkich składników systemu. Kontrola kompletności pozwala na usunięcie z opisu systemu redundancji, synonimów, homonimów, pojęć rozmytych, niedokładnie zdefiniowanych.

Niektóre z metodyk specyfikują większą liczbę kryteriów - w sposób najbardziej pogłębiony zagadnienie to jest traktowane w podejściu EDM. Oryginalną propozycję zawarto w metodyce ISAC, wyszczególniając kryteria funkcjonowalności i wykonalności. Narzędzia wspomaganego komputerem TSI zawierają mechanizmy a u t o m a t y c z n e j k o n t r o l i spełnienia kryteriów oceny. Kontrola ta jest jedną z przesłanek i t e r a c y j n e j realizacji procesu TSI.

[1/] Pakiet wspomagający konstruowanie systemów o nazwie GRAPHDOC. dostosowany do wynagań metodyki ISAC jest w użytkowaniu Instytutu Okrętowego P.G. Istnieją tu też doświadczenia w zakresie wykorzystania systemu PSL/PSA.

10. Synteza porównań

Dotychczasowa analiza porównawcza polegała bądź na opisie wg wybranego wzorca kolejnych metodyk bądź ich porównaniu względem spełnienia poszczególnych kryteriów oceny. Niepełny jest jeszcze uogólniony o b r a z , swoista "mapa terytorium", która w z s y n t e t y z o w a n e j postaci przedstawi metodyki w aspekcie spełnienia wybranych wymogów, potrzeb, oczekiwań oraz stanowiłaby podstawę oceny ich wzajemnych odniesień. Rolę tę pełni tablica 1.

Nie wdając się w bliższą charakterystykę umieszczonych w tym zestawieniu kryteriów /uczyniono to już bowiem w punktach 3 i 4/ poniżej przedstawiono syntetyczne wyjaśnienie użytych w tab. 1 s k r ó t ó w:

- rodzaj metodyki
  - A - akademicka
  - P - profesjonalna
  - P /+A/ - połączenie tych dwu rodzajów z przewagą profesjonalnej.

- wdrożenia
  - W - metodyka zastosowana i zweryfikowana w praktyce,
  - brak wdrożeń /bądź jedynie eksperymentalne, uniwersyteckie/.

- cykl życia systemu
  - P - pełny
  - Z - zbliżony do pełnego
  - O - ograniczony do wycinkowych faz.

- opis statyki / dynamiki dziedziny przedmiotowej 1
  - S - statyka
  - D - dynamika
  - S + D - przewaga opisu elementów statycznych
  - D + S - przewaga opisu elementów dynamicznych.

- teoretyczny model struktur danych
  - OAZ - Obiekt - Atrybut - Związek

Tablica porównawcza metodyk TSI

Tablica 1

Lp	Nazwa metodyki	Rodzaj	Wdro-żenia	Cykl życia	Model danych	Statyka/ dynamika	Zestaw metod i tech.	Komputerowe wspomag.
1	2	3	4	5	6	7	8	9
1	ACM/PCM	A	-	O	OAZ+AD	D+S	P	S
2	CIAM	A	-	O	OAZ+RP/+ WI+RW/	S+D	P	PJ
3	D2S2	P	W	P	OAZ	S+D	V	S/S
4	DADES	A	-	O	R+RP /+RW/	S+D	P	J
5	EDM	A	-	Z	OAZ	S+D	V	-
6	IML	A	-	O	RP+R	S	P	PJ
7	ISAC	P/+A/	W	Z	A	D+S	V	-
8	ISSM	A	-	Z	OAZ	S+D	P	J
9	NIAM	P	-	O	B /+W/	S	P	J
10	REMORA	A	-	Z	OAZ	D+S	P	J
11	SADT	P	W	O	A	S+D	P	S
12	SA-SD	P	W	O	A+R	D+S	V	-
13	SDLA	A	-	O	OAZ	S+D	P	J
14	SYSDOC	P	W	Z	OAZ /+WI+ RW/	S+D	V	S
15	USE	A/+P/	-	P	R+A	S+D	P	S

Źródło: Opracowanie własne

. B - Binarny
. RP - Rachunek Predykatów
. AD - Abstrakcje Danych
. R - model relacyjny
. A - modele własne, autonomiczne
. OAZ + RP - łączenie kilku typów modelu
. /WI/ - model /np. OAZ/ wzbogacony o więzy integralności
. /RW/ - model /np. OAZ/ wzbogacony o reguły wnioskowania.

- zestaw metod i technik dokumentowania /ręcznego/
. U - urozmaicony, bogaty
. P - przeciętny

- narzędzia wspomaganego komputerem TSI
. S - systemy zautomatyzowanego TSI
. J - języki specyfikacji systemu bądź potrzeb
. PJ - projekty języków
. S/S - słowniki/skorowidze danych
. - brak narzędzi.

11. Podsumowanie i wnioski

Niniejsze opracowanie stanowi wynik dłuższych czaso- i pracochłonnych badań nad rozwojem metodyk TSI, poszukiwaniem t r w a ł y c h , chciałoby się powiedzieć naukowych podstaw tej dziedziny. Jest to pierwsza w warunkach krajowych tego typu analiza porównawcza, obejmująca tak szeroką gamę nowoczesnych narzędzi TSI. Jakie wynikają ze zrealizowanych badań wnioski?

W konkluzji przeprowadzonych analiz należy sformułować pogląd, iż n i e m a m e t o d y k, d o g o d n e j d l a w s z y s t k i c h t y p ó w t w o r z o n y c h s y s t e m ó w i n f o r m a t y c z n y c h. Różne metodyki nie są jednakowo dobre dla różnych dziedzin przedmiotowych - każdy typ systemu informatycznego wymaga specyficznego podejścia.
N i e m a w i ę c m e t o d y k i i d e a l n e j mającej przewagę nad pozostałymi - wszystkie mają swoje wady i zalety.

Powyższe stwierdzenia winny stanowić raczej zachętę do głębszego studiowania różnorodnych ujęć metodycznych. Użytkownik, zespół projektowy, w procesie TSI w oparciu o omawiane tu podejścia, ma bowiem do wyboru dwie strategie, z których każda wiąże się z dobrą znajomością stanu wiedzy w tej dziedzinie.
Pierwsza polega na dostosowaniu właściwej metodyki do specyficznej dziedziny przedmiotowej, a druga na stworzeniu własnego podejścia poprzez dobór adekwatnych metod i technik właściwych dla różnych metodyk, dla spójnego rozwiązania systemowego. Pierwsza strategia wymagająca stosowania krok za krokiem wybranego podejścia jest właściwa dla zespołów projektowych o małym doświadczeniu, natomiast drugą zalecić można ekspertom w dziedzinie TSI.

Powyższe wnioski są dość ogólne i w swej wymowie nie są zbyt optymistyczne dla potencjalnych użytkowników metodyk. Należy dodać, iż w badaniach CRIS osiągnięto również bardzo ograniczone cele pragmatyczne. Tym niemniej kilka pragmatycznych wniosków o dość dużym ciężarze gatunkowym dla użytkowników, analityków i projektantów można sformułować. Dotyczą one możliwości realizacji fazy analizy potrzeb w oparciu o metodyki profesjonalne, roli modeli Obiekt-Atrybut-Związek oraz funkcji narzędzi wspomagających proces TSI.

Wydaje się, iż istnieje aktualnie kilka metodyk, chodzi głównie o ISAC, SA-SD i SADT, które mogą okazać się niezmiernie użytecznymi narzędziami od zaraz w zespołach twórców systemów do przeprowadzenia fazy analizy potrzeb. Wymaga to oczywiście odpowiedniej akcji szkoleniowej lecz podejścia te są wielokrotnie zweryfikowane empirycznie. Posiadają one formalizmy i walor wspólnego języka pomiędzy użytkownikami a informatykami. Działania w tym kierunku winny opierać się na bardziej pogłębionych materiałach źródłowych, dobrej znajomości metod i technik właściwych dla każdego podejścia.

Częstą podstawą modelowania dziedziny przedmiotowej jest model Obiekt-Atrybut-Związek powiązany z modelowaniem procesów przy użyciu kategorii funkcji, zadań i operacji. Podejście to w sposób pełny, praktyczny i przystępny dla nieinformatyków pozwala zrealizować fazę modelowania koncepcjualnego baz danych.

Dość ostrożnie należy traktować kwestie w s p o m a g a n e g o k o m p u t e r e m TSI. Wytworzone narzędzia są jeszcze niezupełnie dojrzałe. Są one niezmiernie istotnym przyszłościowym składnikiem środowiska tworzenia systemów informatycznych lecz na obecnym etapie jeszcze niedoskonałym. Równie znikoma jest ich dostępność na rynku informatycznym, zwłaszcza krajowym. Jednak kwestia podejmowania intensywnych prac jest aktualnym wyzwaniem badawczym.

B i b l i o g r a f i a

/BRAN 83 / Brandt I., Comparative Study of Information Systems Design Methodologies, w OLLE 83, ss. 9-36 .

/CERI 86 / Ceri S. /ed./, Requirements Collection and Analysis in Information Systems Design, w KUGL 86 , ss. 205-217 .

/CERI 83 / Ceri S. /ed./, Methodology and Tools for Data Base Design, North Holland, Amsterdam 1983.

/FRY 78 / Fry J.P., Teorey T.J., Oberlander L.B., Database Design Tools and Methods. A Survey and Tutorial, Technical Report 78 DB 14, The University of Michigan, Ann Arbor 1978.

/KUGL 86 / Kugler H.J. /ed./., Information Processing 86, North Holland, Amsterdam 1986.

/LOCK 86 / Lockeman P.C., Mayr H.C., Information Systems Design: Techniques and Software Support, w KUGL 86 , ss. 617-634 .

/MADD 83 / Maddison R.N. /ed./., Information Systems Methodologies, Wiley Heyden London 1979.

/NIED 77 / Niedzielska E. /red./., Projektowanie systemów informatycznych, PWE, Warszawa 1977.

/OLLE 82 / Olle T.W., Sol H.G., Verrijn Stuart A.A. /eds./, Information Systems Design Methodologies: A Comparative Review, North Holland 1982.

/OLLE 83 / Olle T.W., Sol H.G., Tully C.J. /eds./., Information Systems Design Methodologies, North Holland, Amsterdam 1983.

/TEIC 77 / Teichrew D., Hershey E.A. III, PSL/PSA: A Computer - Aided Technique for Structured Documentation and Analysis of Information Processing Systems, IEEE Transactions on Software Engineering, 1/1977, vol. SE-3, ss. 41-48.

/WRYC 85/ Wrycza S., Konceptualne modelowanie baz danych organizacji gospodarczych. Rozprawy i Monografie UG, Gdańsk 1985.

/WRYC 86/ Wrycza S., Bibliografia tworzenia systemów informatycznych, Instytut Okrętowy PG, Gdańsk, 2479/86, 1986.

/WRYC 87a/ Wrycza S., Tworzenie systemów informatycznych w oparciu o model Obiekt-Atrybut-Relacja, Instytut Okrętowy PG, Gdańsk 1/1987.

/WRYC 87b/ Wrycza S., Metodyka ISAC - założenia, cele, techniki, zastosowania, Instytut Okrętowy PG, Gdańsk 2/1987.

/WRYC 87c/ Wrycza S. Analiza porównawcza metodyk tworzenia systemów informatycznych. Prace Badawcze IO PG, Gdańsk 1987.

Stanisław Bała
Gdańsk

E U R O G R A P H I C S' 87

informacja o kongresie grafiki komputerowej

WSTĘP

Stowarzyszenie EUROGRAPHICS, (The European Association for Computer Graphics), zostało założone w 1980r w celu rozpowszechniania i rozwoju grafiki komputerowej w Europie. Organizacja ta rozwinęła się gwałtownie wraz ze wzrostem znaczenia komputeryzacji w nauce, technice i innych dziedzinach życia i z rozpowszechnieniem mikrokomputerów.

Corocznie EUROGRAPHICS organizuje międzynarodowy kongres będący najważniejszym spotkaniem stowarzyszenia i jednocześnie najodpowiedniejszą okazją do wymiany informacji na temat osiągnięć i tendencji rozwoju grafiki komputerowej. EUROGRAPHICS'87, ósmy kongres miał miejsce w Amsterdamie od 24 do 28 sierpnia 1987r. Współorganizatorem kongresu było Holenderskie Centrum Matematyki i Informatyki (C.W.I.).

Program kongresu był bardzo obszerny i obejmował kursy szkoleniowe, konferencję, przemysłową wystawę sprzętu i oprogramowania z seminariami wystawców oraz konkursy sztuki komputerowej w kategoriach grafiki i filmu. Przez dwa dni poprzedzające konferencję odbywały się płatne kursy szkoleniowe. Dwadzieścia pięć firm z siedmiu krajów pokazywało swoje urządzenia i oprogramowanie na wystawie przemysłowej. Do konkursów wystawiono dwadzieścia trzy obrazy i dwadzieścia osiem filmów animowanych kom-

puterowo. W kongresie wzięło udział czterystu czterdziestu jeden uczestników z dwudziestu siedmiu krajów, w tym jeden z Polski. Oficjalnym językiem kursów i konferencji był angielski.

KURSY SZKOLENIOWE

Przed konferencją odbyło się trzynaście kursów prowadzonych przez międzynarodowych uznanych ekspertów grafiki komputerowej i innych odnośnych dziedzin. Kursy te były zaprojektowane dla szerokiego audytorium; niektóre jako wprowadzające dla początkujących użytkowników grafiki, inne dla pogłębienia wiedzy o rozwijających się dziedzinach zastosowań lub też popularyzacji nowych przedmiotów

Oto tematy kursów:

- Wprowadzenie do CAD
- Inżynierskie bazy danych
- Zastosowania VLSI w modelowaniu przestrzennym
- Konstruowanie objektów przestrzennych
- Poligrafia mikrokomputerowa
- Przegląd grafiki do animacji tekstów
- Wymiana danych w CAD
- Urządzenia grafiki komputerowej
- Modelowanie geometryczne
- Komputer narzędziem artysty i projektanta
- Grafika zorientowana objektowo
- Systemy komunikacji człowiek - maszyna

Kursy zastosowań grafiki komputerowej do projektowania (Computer Aided Design), modelowanie geometryczne i popularyzacja rozwoju urządzeń komputerowych mają już

tradycyjnie duże powodzenie na kongresach EUROGRAPHICS. Problematyka standaryzacji i obsługi baz danych systemów CAD/CAM jest konsekwencją postępującej integracji systemów użytkowych. Nowe dziedziny zastosowań, jak poligrafia, animacja w różnych specjalnościach projektowania, reklamy handlowej, wreszcie kinematografii i sztuki rozwijają się szybko i cieszą się dużym zainteresowaniem.

PROGRAM KONFERENCJI

Konferencja obejmowała sesje plenarne, na których występowali zaproszeni prelegenci i równoległe problemowe sesje referatów, dyskusji panelowych i komunikatów. Jednocześnie odbywały się seminaria przemysłowe, wystawa sprzętu i oprogramowania oraz pokazy filmów i utworów sztuki komputerowej. Tematyka sesji plenarnych, paneli i referatów oddaje ogólną problematykę konferencji i zarazem aktualny zakres zastosowań grafiki komputerowej.

Sesje plenarne:

- Wpływ telewizjii wielkoobrazowej na grafikę komput.
- Obraz graficzny w umyśle człowieka i w komputerze
- Wzornictwo przemysłowe a projektowanie wspomagane
- Grafika komputerowa w latach 1970 - 1990
- Nowy papirus dyski laserowe (CD-ROM)

Sesje panelowe:

- Inteligentne systemy CAD
- Okienkowe zarządzanie wyższego poziomu
- Zastosowania grafiki w projektowaniu
- CAD/CAM w małych i średnich przedsiębiorstwach
- Badania i rozwój grafiki komputerowej

- VLSI dla grafiki (Very Large Scale Integration)
- Ocena i certyfikacja realizacji GKS

W grupach tematycznych wymienionych niżej wygłoszono ponad czerdzieści referatów:

- Modelowanie gepmetryczne
- Zastosowania systemów grafiki
- Synteza obrazu
- Procesy równoległe w grafice
- Zastosowania w medycynie
- Wyposażenie
- Grafika zorientowana obiektowo
- Zastosowania grafiki w projektowaniu
- Interakcja
- Wizualizacja objektów trójwymiarowych
- Geometria obliczeniowa
- Projektowanie wspomagane komputerowo
- Normalizacja

Komunikaty:

W celu przyśpieszenia wymiany informacji zorganizowano specjalne sesje komunikatów aby umożliwić dyskusje nad inowacyjnymi pomysłami lub zastosowaniami, zanim zostaną skrystalizowane i opracowane w formie gotowych publikacji. Zgłoszono osiemnaście komunikatów, po dziewięć w dwóch dziedzinach: zastosowań i narzędzi. W większości tyczyły tematów badawczych, metod lub systemów oprogramowania nietowarowych lub będących w toku opracowania. Między innymi prezentowano ciekawy amerykański system AGIP (A Graphical Interaction Package), stworzony według bardzo nowoczesnego modelu komunikacji z użytkownikiem w interaktywnym środowisku graficznym.

WYSTAWA PRZEMYSŁOWA

Na wystawie obecni byli producenci znani na rynku sprzętu komputerowego, tacy jak Prime Computer, czy Honeywell oraz specjalizujący się w urządzeniach graficznych, jak Apollo. Większość wystawców stanowiły jednak różnej skali firmy zajmujące się wytwarzaniem oprogramowania lub kompletowaniem systemów graficznych. Niektóre z nich prezentowały bardzo obszerny i bogaty dorobek włącznie z własnymi opracowaniami o charakterze klasyfikacji i standaryzacji (np GRAL & Partners).

Poza tymi dwiema grupami firm rynkowych na wystawie były też obecne organizacje typu 'non profit', jak EUROGRAPHICS, czy C.W.I. (Centrum voor Wiskunde en Informatica) oraz poważne firmy zajmujące się publikacją czasopism i książek naukowych i technicznych, takie jak Harcourt Brace Janovich Ltd, czy Springer Verlag GmbH.

SPOSTRZEŻENIA

Gwałtowny rozwój techniki komputerowej i masowe rozpowszechnienie minikomputerów i mikrokomputerów w ostatnich latach wywołuje szybki postęp w rozwoju oprogramowania i szerokiego stosowania komputerów we wszystkich dziedzinach ludzkiej działalności.

Grafika komputerowa okazuje się najbardziej dogodnym i obiecującym sposobem komunikacji człowieka z maszyną. Najnowsze osiągnięcia w wytwarzaniu stacji graficznych pozwalają już na modelowanie przestrzennych obiektów ze znakomitą symulacją rzeczywistości, a także na ich animację w czasie rzeczywistym.

Obok wytwarzania i przetwarzania obrazów graficznych w komputerze rozwinęły się również bardzo zaawansowane techniki przenoszenia obrazów do komputera i reprodukcji obrazów komputerowych. Do szybkiego kopiowania obrazów wielobarwnych dostępne są doskonałe plotery strumieniowe i kopiarki termiczne używające barwne folie jednorazowe.

Interaktywne systemy wspomagające projektowanie i produkcję (CAD/CAM) znalazły powszechne zastosowanie w technice i przemyśle. Są to zwykle kompleksowe systemy bogato wyposażone funkcjonalnie, przeważnie wyspecjalizowane w konkretnych zastosowaniach. Miarą ich nowoczesności staje się technika komunikacji człowieka-maszyna (user interface).

Profesjonalne systemy graficzne posługują się urządzeniami wysokiej jakości, tj szybkimi minikomputerami 32-bitowymi i specjalnymi procesorami graficznymi oraz bogatym zastawem peryferiów z inteligencją. Jako typowy przykład, jeden z wielu, może posłużyć Apollo DN 590.

Mikrokomputery 16-bitowe można spotkać jedynie w ograniczonych zastosowaniach szczególnych, np do interpretacji zdjęć rentgenowskich, jako terminale do zarządzania i przygotowania danych, lub jako stacje graficzne wyposażone dodatkowo w procesor graficzny i odpowiedni monitor (patrz procesor LogiGraphics RGS-5640).

Punkt ciężkości zainteresowań wytwórców oprogramowania przesuwa się na nowe tematy, takie jak graficzne modelowanie, symulacja, animacja, sztuczna inteligencja. Wśród nowych i żądnych dziedzin zastosowań grafiki trzeba wymienić poligrafię, reklamę handlową, telewizję i kinematografię. Widać też ciekawe prace nad przekazem obrazów bitowych na potrzeby dziennikarskie (electronic journal)

oraz w zakresie generacji fraktali w przestrzeni dwu i trójwymiarowej.

W ślad za owocnymi pracami nad międzynarodowym standartem grafiki komputerowej GKS (Graphical Kernel System), kontynuowane są obecnie intensywne prace nad innymi normami w tej dziedzinie. Nowe urządzenia graficzne są już wyposażane zgodnie ze standartami. Ponadto szereg instytucji zajmujących się oprogramowaniem oferuje i wykonuje atesty oprogramowania na zgodność ze standartami (patrz załączone ogłoszenie brytyjskiego NCC).

NCC THE NATIONAL CENTRE FOR INFORMATION TECHNOLOGY

-80-

Testing Services

The GKS Test Service

The GKS Test Service is based on the technical specifications contained in ISO (International Standards Organisation) 7942-1985 Information processing systems — Computer graphics — Graphical Kernel System (GKS) functional description and ISO DIS 8651 Information processing systems — Computer Graphics — Graphical Kernel System (GKS) Language Bindings; Part 1: FORTRAN. It consists of test programs, initialisation programs and utility routines, each of which is a FORTRAN source program. Supporting documents and procedures indicate the expected results for each test program.

The testing of a GKS implementation determines the conformance of the product to ISO 7942-1985, the GKS standard. GKS testing will contribute to the development of products which are in conformance with the standard and this should, in turn, provide the following benefits:

- **allow application programs using GKS to be easily portable between different installations.**
- **aid the understanding and use of graphics methods by application programmers.**
- **provide a wider market for graphics products.**
- **result in less re-training and re-programming being required and hence reduced maintenance costs.**

Background

GKS provides a basic graphics system for the display and manipulation of pictures. It handles only two dimensional graphical data, but an extension of the standard currently under development, called GKS-3D, will provide facilities for processing three dimensional data.

GKS is defined independently of a programming language; language bindings (mappings of the GKS functional specification to the constructs of different programming languages) are the subject of a separate standard. Bindings are under development for FORTRAN, Ada,* Pascal and 'C'.

The GKS standard is designed in a level structure, the nine levels, which are upwards compatible from level 0a to level 2c, provide for implementations of varying capabilities.

The relevance of GKS conformity testing was appreciated at an early stage in the GKS standardisation process. The Commission of the European Communities (CEC) played a significant role in funding collaborative workshops, where technical experts from other areas of software testing met with GKS experts to design a feasible strategy for testing implementations of GKS. As a result of these CEC workshops, GKS test software implemented in FORTRAN was developed in parallel with the standardisation of GKS itself. This project was supported initially by the CEC and the Science and Engineering Research Council (SERC) in the United Kingdom and later by Gesellschaft fuer Mathematik und Datenverarbeitung mbH (GMD) in FR Germany. The University of Leicester, Technische Hochschule Darmstadt and GMD produced test software, for testing level 2b GKS implementations with a FORTRAN binding. Recent CEC funding has enabled the early software to be updated and integrated into an international test service.

The test software is being converted to the 'C' language and a service for testing GKS implementations which use the 'C' language binding will be available in the future.

Description of Testing Strategy

The test strategy employed in the GKS test software uses the falsification approach. This method attempts to prove that an implementation is incorrect by running a set of test programs and comparing the results with expected results. Falsification cannot prove the correctness of a GKS implementation, however it does give confidence that an implementation which passes all the tests provides an accurate interpretation of the standard.

The test strategy involves five distinct test series:

- data consistency test series
- data structure test series
- error handling test series
- input/output test series
- metafile test series

The data consistency test series examine the GKS description tables. The tests check the values in the description tables for consistency and conformity with the GKS standard. Details of the results in the form of a partial test report are produced automatically.

The data structure test series ensure that the values in the GKS state lists are manipulated correctly, by setting, modifying and inquiring the values. Details of the results in the form of a partial test report are produced automatically by each of the test programs.

*Ada is a registered trademark of the United States Government (Ada Joint Program Office).

1966-1987 NCC 21 Years of Service

The National Computing Centre Limited Oxford Road Manchester M1 7ED
Telephone 061-228 6333 Telex 668962

The error handling test series produce error situations and then check that an error mechanism in line with the GKS standard is supported by the implementation. Each possible error that can be generated for each GKS function is tested. Details of the results in the form of a partial test report are produced automatically by each of the test programs.

The input/output test series provide a check of the GKS System as a whole, through a comprehensive set of tests which covers all the input and output capabilities of GKS. Graphical displays are produced by the tests and checked against a set of reference pictures. The GKS Evaluator's Manual gives a list of items to be checked for each picture. Input is tested by a set of defined operator actions which should produce specific results on a display.

The metafile test series check that the GKS metafile is used correctly. Metafiles are created and checkpoints are used to enable visual comparison between screen output and reference pictures. The metafiles are then interpreted and the sessions are interrupted at exactly the same check-points so that the output from metafile interpretation can be checked against the reference pictures.

Overview of the Testing Process

The Testing Process for a GKS implementation consists of a number of stages:

— initial enquiry about testing

— delivery of the testing tools

— pre-testing

— on-site testing

— preparation of report

— issue of certificate (where appropriate)

The Pre-testing stage begins when a customer requests testing of a GKS implementation from a Testing Laboratory. At this stage the customer prepares his implementation for testing by running the GKS test software on it.

The on-site testing stage is the formal testing of the GKS implementation, which is normally carried out at the customer's site under the supervision of a member of staff from the Testing Laboratory.

The final stage is the preparation of the report which gives details of the results of testing for the implementation and configuration under test. A certificate may be issued if the requirements of the certification scheme have been met.

A list of tested GKS implementations is published on a quarterly basis.

Required Resources

The following resources are required in the computer system in order to test the GKS implementation with the GKS test suite:

— ISO standard Fortran 77 compiler to compile the test programs (preferably certified)

— a magnetic tape drive or other input or telecommunications means acceptable to the Testing Laboratory

— a printer for printing the test results

— one or more workstations.

— sufficient storage to compile, link and execute the test programs.

Availability of GKS Test Service

At present, GKS Testing is only available for GKS implementations which use the FORTRAN language binding. Test suites, which use the 'C' and Ada* language bindings are under development and information on the status of these can be obtained from any of the Testing Laboratories.

For further information on the GKS Test Service please contact the Testing Services department at NCC.

NCC is the UK Centre for Informatio Technology. Backed by, and in co-oper with, government, the IT industry, and I NCC directs technical programme administers national schemes, and de products and services to promote the c use of Information Technology.

Membership

Any organisation can benefit from NCC membership or, for overseas organisations, affiliate membership. It gives them a direct link to NCC's products, services, and experience, and the opportunity to participate in discussion and working groups which influence NCC and government policy.

The National Computing Centre Limited

Head Office
Oxford Road, Manchester M1 7ED
Tel: 061-228 6333 Telex: 668962
Telecom Gold 81: NCT 008
Fax (Groups 2 & 3): 061 228 2579

London and South East
11 New Fetter Lane, London EC4A 1PU
Tel: 01-353 4875
Telecom Gold 81: NCT 011
Fax (Groups 2 & 3): 01 583 8275

Midlands
7th Floor, Devonshire House,
Great Charles Street, Birmingham B3 2PL
Tel: 021-236 6283
Telecom Gold 81: NCT 016

North and North Wales
Bracken House,
Charles Street, Manchester M1 7BD
Tel: 061-228 6333

Northern Ireland
117 Lisburn Road, Belfast BT9 7BP
Tel: 0232 665997
Telecom Gold 81: NCT 018

Scotland
2nd Floor, Anderston House,
389 Argyle Street, Glasgow G2 8LR
Tel: 041-204 1101
Telecom Gold 81: NCT 017

South West and South Wales
3rd Floor, The Crescent Centre,
Temple Back, Bristol BS1 6EZ
Tel: 0272 277077
Telecom Gold 81: NCT 012

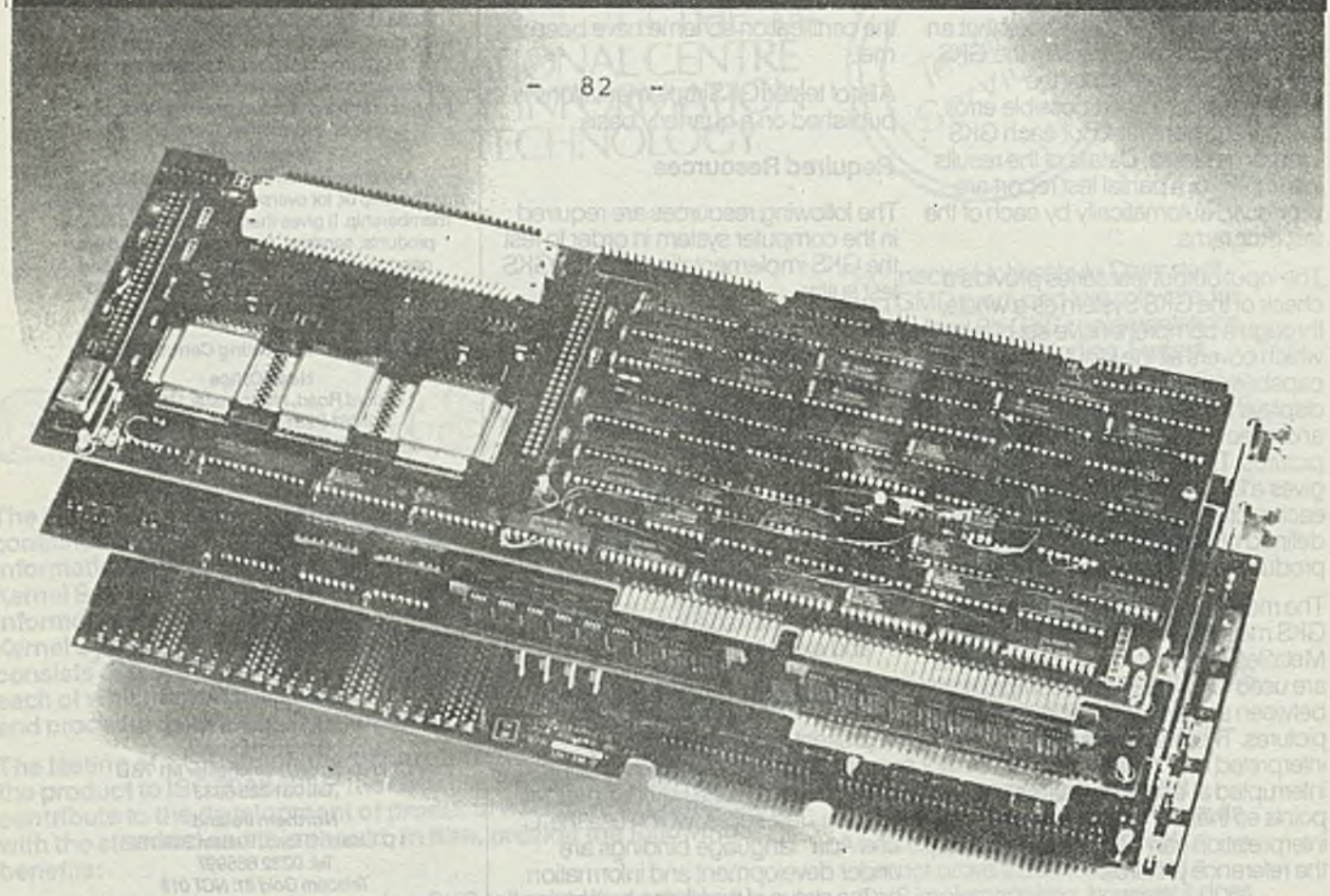
This publication is issued subject to alteration or withdrawal without notice. NCC prices and terms and conditions of business are available at NCC Regional Offices and International Distributors.

NCC THE NATIONAL CENTRE FOR INFORMATION TECHNOLOGY

UNITED KINGDOM

© The National Computing Centre Limited 1987
P2000 Printed in England 1/8/87 27X1036

RGS-5640 5 MIPS + REAL TIME PREVIEW = POWER PERFORMANCE



- 33 MHz Fairchild Clipper
- 32-Bit VLSI Draw Processor
- 8-Bit Double Buffer Mode or
- 24-Bit Single Buffer Mode
- 16-Bit Z-Buffer
- 32-Bit Graphics Bus Interface
- 16.7 Million Colors
- UNIX System V
- Demand-Paged Virtual Memory

- NTSC/PAL Compatible RGB Output
- GENLOCK to Composite Sync
- Real-Time Motion Preview
- 3-D Solid Modeling
- Hidden Surface Removal
- Gouraud Shading
- Light Source Modeling
- Shadowing and Texture Mapping
- Perspective

ANIMATION

Animation requires graphics performance unlike any other application. First, you must be able to create fully-rendered images. Then you need to be able to easily send them off to film or tape. What we offer beyond this bare minimum is the ability to preview your scenes in real time. When it comes to story-boarding for concept approval, we provide you the ability to present choreographed scenes — and the ability to modify your scene interactively!!!

SOLID MODELING

Modeling has been performed within the PC environment for years, but it has been generally limited to 2-D static images. We provide you the flexibility to present not only 2-D static wireframes, but also 3-D dynamic solids. Develop your object as it will really appear as a finished product. Then manipulate it like you've never been able to before. Perform cross-sections or extrusions. Apply a colored or textured surface. Maneuver it in 3-D space. Solid modeling for solid ideas.

LogiGraphics, Inc.

Product Specification Summary RGS-5640

HARDWARE:

Graphics Processor —

Fairchild Clipper @ 33 MHz
4 or 8 MB Processor RAM
16-Bit IBM PC/AT Bus Interface
32-Bit Graphics Bus Interface

Frame Buffer —

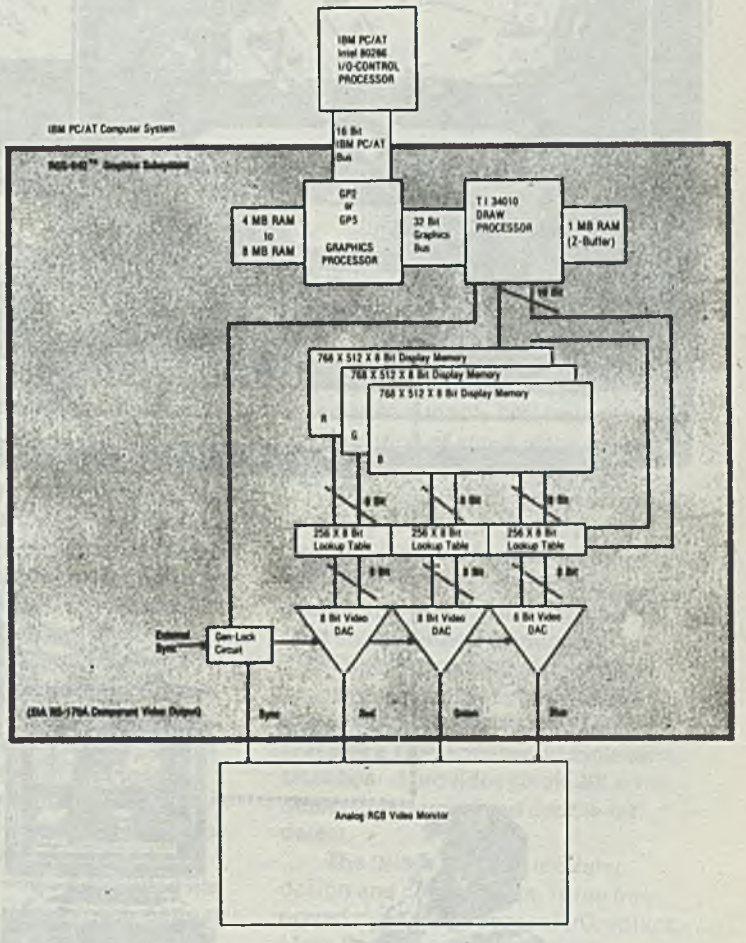
IMAGE STORAGE —
768 X 512 24-Bit Image Buffer (768 X 768 Europe)
640 X 485 Display Resolution (768 X 575 Europe)

DRAW PROCESSOR —
Texas Instruments 34010 Draw Processor
1 MB Processor RAM (Includes Display Resolution Z-Buffer)
32-Bit Graphics Bus Interface

VIDEO INTERFACE —
Triple 8-Bit Video DACs
Genlock to RS-170 Composite Sync Signal
RS-170A Component RGB Outout Signals
30 Hz Interlaced 640 X 485 (NTSC)
25 Hz Interlaced 768 X 575 (PAL)

COLOR RANGE —
Color Map Mode — 256 Colors (Double Buffered)
Displayable from a 16.7 Million Color Palette
RGB Mode — 16.7 Million Colors Displayable

RGS-640™ Graphics Subsystem Block Diagram



SOFTWARE

Graphics Processor —

UNIX System V Operating System
C Compiler, FORTRAN 77
Graphics Interface Library, including:
- *Hierarchical Display List*
- *3-D Graphics Primitives*
- *3-D Modeling Support Routines*
- *Text*
- *Raster Operations*
- *Viewing Control*
- *Multiple-Windows*
- *Input Device Drivers*

Draw Processor —

Primitive Rendering Utilities
Clipping Processing
Z-Buffer Management

LogiGraphics, Inc.

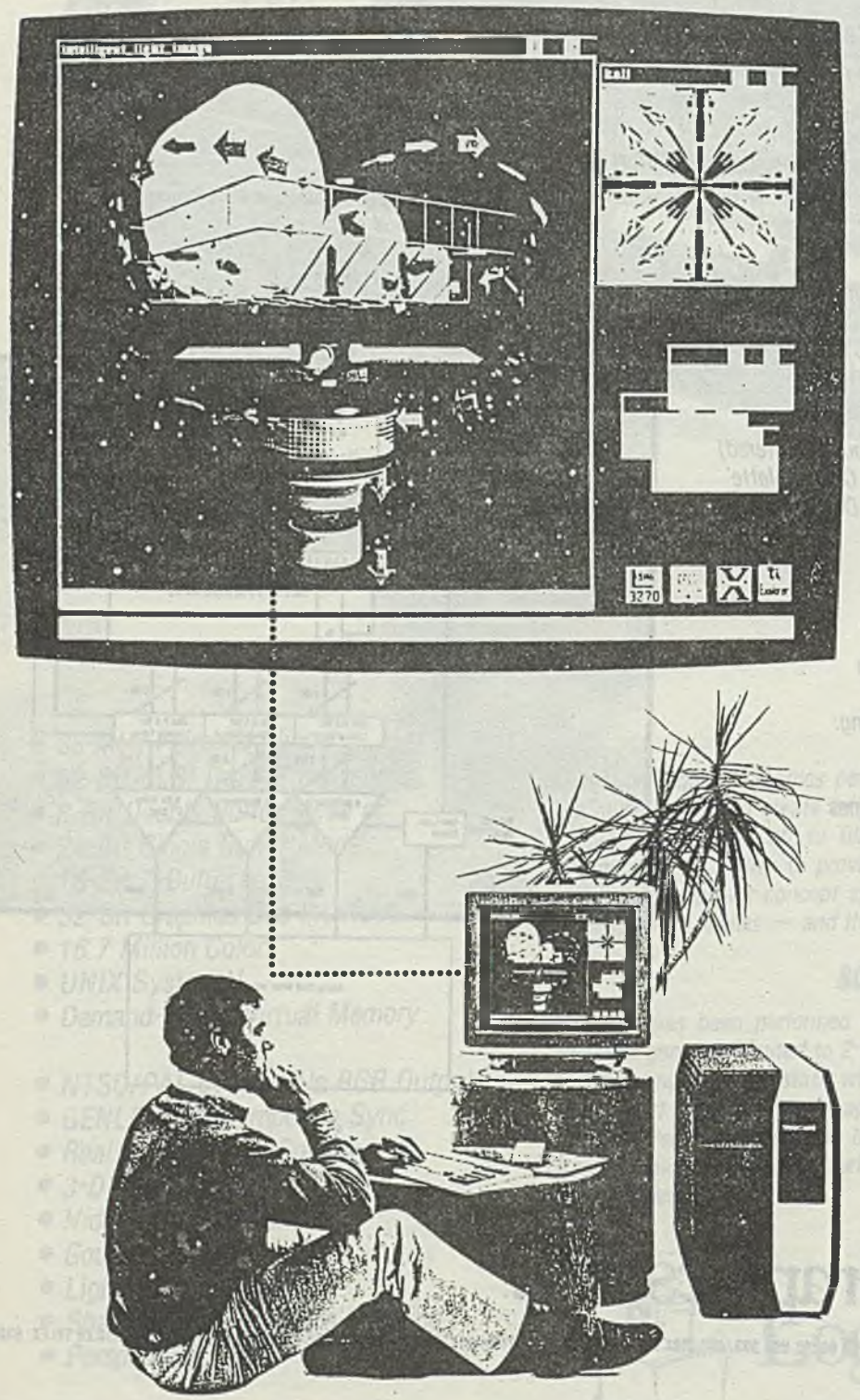
1050 Walnut, Suite 325, Boulder, CO 80302 USA 303/443-0191 TELEX: 988497 126 Avenue Victor Hugo, 92100 Boulogne/Seine, France (1) 48.25.13.25 TELEX: 632137 F

UNIX is a Trademark of AT&T Bell Labs IBM PC/AT is a Trademark of International Business Machines Corp. Renaissance Graphics Systems is a Trademark of LogiGraphics Inc. Clipper is a Trademark of Fairchild Semiconductor Corporation

DOMAIN WORKSTATIONS

DN590 TURBO

apollo



Description

The DN590 TURBO is Apollo's high-end color graphics workstation designed specifically for compute intensive applications that require shaded 3-D graphics. The DN590 TURBO combines the dedicated computing power of the MC68020 microprocessor and the MC68881 floating point coprocessor with a series of highly specialized graphics processors. The result is a uniquely well balanced graphics workstation fully integrated within Apollo's family of workstations.

The DN590 TURBO offers truly spectacular true color graphics. A dedicated video memory offers 24 color planes and the ability to display over 16.7 million colors simultaneously in a window. For high speed 3-D rendering the DN590 TURBO offers both double buffering and Z-buffering operations. The optional 3DGA™ graphics accelerator not only transforms and clips 3-D floating point coordinates, but performs lighting model calculations at extremely fast rates. In addition, the DN590 TURBO has virtual memory graphics that lets users handle the largest and most complex graphics application without display list size limitations.

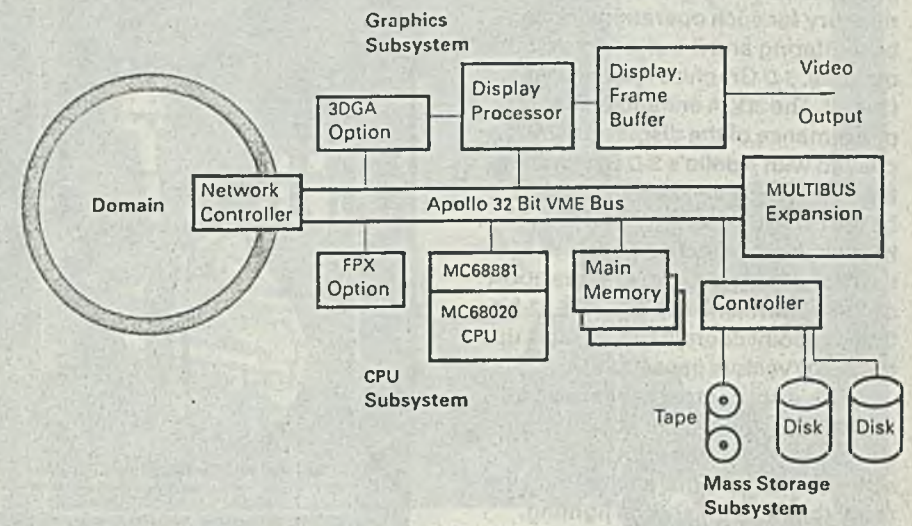
As a fully compatible member of Apollo's family of workstations, the DN590 TURBO lets users enjoy the high productivity of distributed processing afforded by the Domain computing environment.

Features

- Double buffering of pseudo color images, with full Z-buffer support
- Advanced lighting model including: up to 16 light sources, 24 plane RBG support, colored light sources, attenuation of light sources, spotlights with, attenuation, specularity, highlights, and surface properties

- 16.7 million colors displayed, from a palette of 16.7 million colors; 24 color bit-planes planes
- Over 130,000 3-D floating point transformed, clip-checked vectors per second with 3DGA
- Extremely fast, smooth shaded Z-buffered polygons
- 19-inch non-glare monitor; high-resolution 1280 x 1024, bit-mapped display
- Integrated MC68020 processor and MC68881 floating point coprocessor with 16K byte physical cache memory
- Up to 56 concurrent processes; 2G bytes virtual address space per process
- Optional high-performance floating point coprocessor accelerator (FPX™)
- 5¼ inch 190M byte formatted and dual 190M byte Winchester disk options; ¼ inch 60M byte streaming tape drive option
- Four-slot IEEE 796 MULTIBUS adaptor options

DN590 TURBO's Balanced Architecture



Benefits

- High performance compute-intensive capacity coupled with tightly integrated graphics provide a well balanced workstation environment
- Integrated graphics processors yield highest levels of interactive performance in a sophisticated 3-D multiwindow environment
- Double buffering provides smooth dynamic motion of wireframe and solid geometric objects
- Sophisticated lighting model for fast realistic geometry at render time
- 3DGA module provides hardware support for lighting model calculations
- Virtual memory graphics allows rapid display of objects of almost unlimited size without the limitations of dedicated display list memory
- Multiple light sources permit greater control of computer synthesized renderings
- Compatible with entire Apollo product family

Advanced Functionality

The DN590 TURBO's computational power, and high resolution graphics are displayed in a true multiwindow, multitasking environment. Apollo's low-profile, detachable keyboard comes with programmable keypad and optional mouse.

Based on the powerful MC68020 microprocessor and the MC68881 floating point coprocessor, the DN590 TURBO also features an optional high performance FPX floating point accelerator. And the DN590 TURBO features 2 gigabytes of virtual address space per process so that users can develop and run extremely large programs.

Apollo supports the two most popular UNIX® operating systems: Berkeley 4.2 and System V Release 2. Users can run applications in either operating system or both simultaneously. Users can enjoy all the benefits of the UNIX standards running in a distributed processing environment.

The AEGIS™ operating system provides a true multiwindow, multitasking environment and a distributed file system that lets users transparently share data and resources. Any of these operating systems can run in different windows simultaneously, letting users easily work in the operating system they select.

In addition the DN590 TURBO takes full advantage of Apollo's Public Domain™ networking, linking workstation networks to a host of other computing environments such as PC, mainframe, superminis and a host of UNIX-based systems.

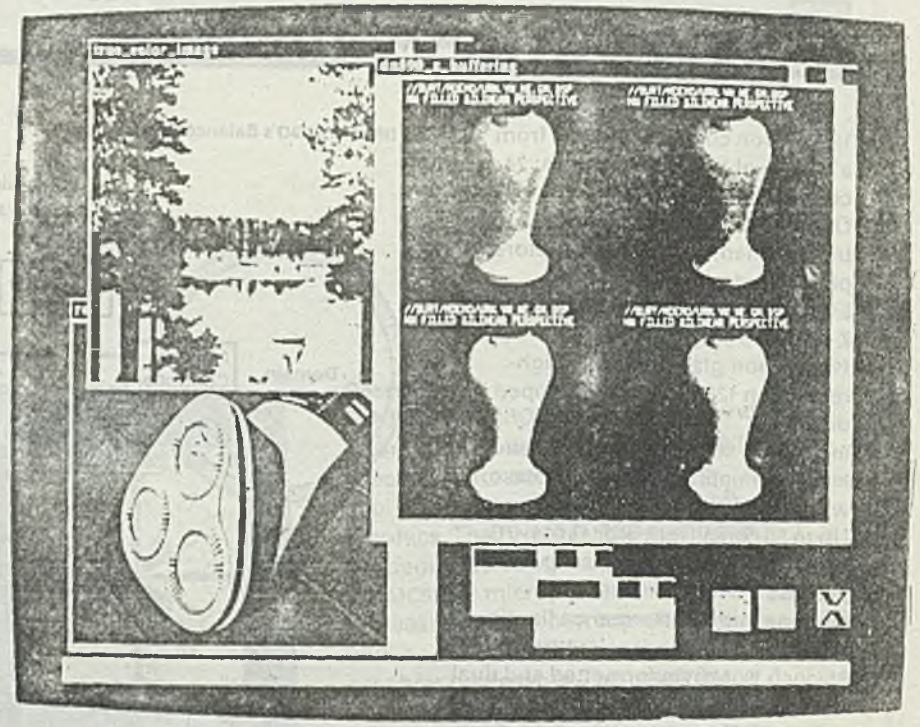
Advanced System Architecture

The DN590 TURBO's CPU features a combination of some of the industry's most advanced technologies.

The high-speed, 16K byte cache memory is fully integrated into the pipelined CPU architecture. This physical cache memory lets programs access instructions and data in a no-wait state implementation.

The DN590 TURBO also provides access to up to 16M bytes of high-speed ECC memory. This ECC memory features a 120-nanosecond cycle time DRAMs, and provides single-bit error detect-and-correct and double-bit detect.

The DN590 features modular design and construction. Three integrated modules – Processor/Graphics, Mass Storage, and MULTIBUS™ modules – fit neatly into one system unit. Modules feature separate fans and power supplies that lend themselves to high reliability and makes servicing easy.



Graphics Subsystem
The graphics section is the key to the DN590 TURBO's virtual memory graphics capabilities. It includes a display processor, and integral image display memory for such operations as double buffering and Z-buffering, and the optional, 3-D Graphics Accelerator (3DGA). The 3DGA enhances the performance of the display of objects created with Apollo's 3-D Graphics Metafile Resource (GMR™), which is based on The Programmer's Hierarchical Interactive Graphics System (PHIGS) constructs. And the 3DGA can transform and clip 3-D, IEEE 32-bit floating point coordinates at rates up to 130,000 vectors per second.

The display processor supports the DN590's 19-inch, 1280 x 1024 color display that features both tilt and swivel, and a non-glare filter that eliminates the need for special lighting. The display processor contains multiple bipolar drawing processors, dedicated CMOS VLSI draw processors, 6M bytes of dedicated display memory, and the video logic required to directly drive the color monitor. The graphics subsystem performs basic functions such as bit-blts, raster operation, drawing and fill operations. And it performs all high-level functions such as complex light source calculations, Gouraud shading, 3-D matrix transforms, and clipping.

The DN590 TURBO's combination of functionality and performance makes it ideally suited for a wide range of graphics intense applications such as 3-D solids modeling, wireframe and image processing. Vector generation is at a rate in excess of 16M pixels/second. Rectangular areas can be filled at a rate in excess of 125M pixels/second. Block transfers (bit-blts) are performed at a rate of 22M pixels/second.

The display memory contains on-board microcode and custom gate array logic to directly support bit-blts, fills, and raster operations. Video RAM technology is employed to provide a 2048 x 1024 x 24-plane display memory that features high drawing bandwidth due to the parallel random access and video refresh paths.

The display frame buffer memory is designed to allow advanced operations such as double buffering, and Z-buffering to operate at very high speeds. In addition the 3DGA aids in this process by performing the appropriate transformations. The 3DGA dedicated graphics processor is a multistage processing system that handles all the complex geometry features required for real-time graphics operations. Through the use of a special VLSI processor and an on-board MMU, the 3DGA can actually process complex objects directly from Graphics Metafiles resident in virtual memory. The 3DGA provides a significant technology breakthrough by bringing virtual memory techniques to graphics processing. This allows even very large objects to be rapidly displayed without regard for any limited display list memory.

Once the Metafile is read, the 3DGA traverses the hierarchical structures as required by the selected view, while simultaneously performing bounding box trivial rejection. 3DGA then interprets and executes the commands, attributes, and control information that represent the geometry of the stored object. The 3DGA transforms the three-dimensional floating point coordinates in orthographic or perspective projection using special VLSI floating point processors and performs hardware clipping prior to output of the resulting screen space primitives to the display processor for drawing.

Advanced Lighting Model
The DN590 TURBO features the most advanced lighting model available on any workstation. Multiple light sources are incorporated to permit greater control of computer synthesized images. Up to 16 light sources can be manipulated with features such as:

- *Direction*
- *Position*
- *Color*
- *Ambient lighting*
- *Concentration*
- *Attenuation*
- *Cone of influence (Spotlights)*

In addition the DN590 TURBO's lighting model allows object surfaces to have distinct properties such as specular hue, base hue, and reflection coefficients, to define and simulate the surface characteristics of many materials. The lighting model is greatly enhanced through the use of the 3DGMR software libraries. Apollo's enhanced Phong lighting model is used by the 3DGMR to provide realistic computer generated imagery.

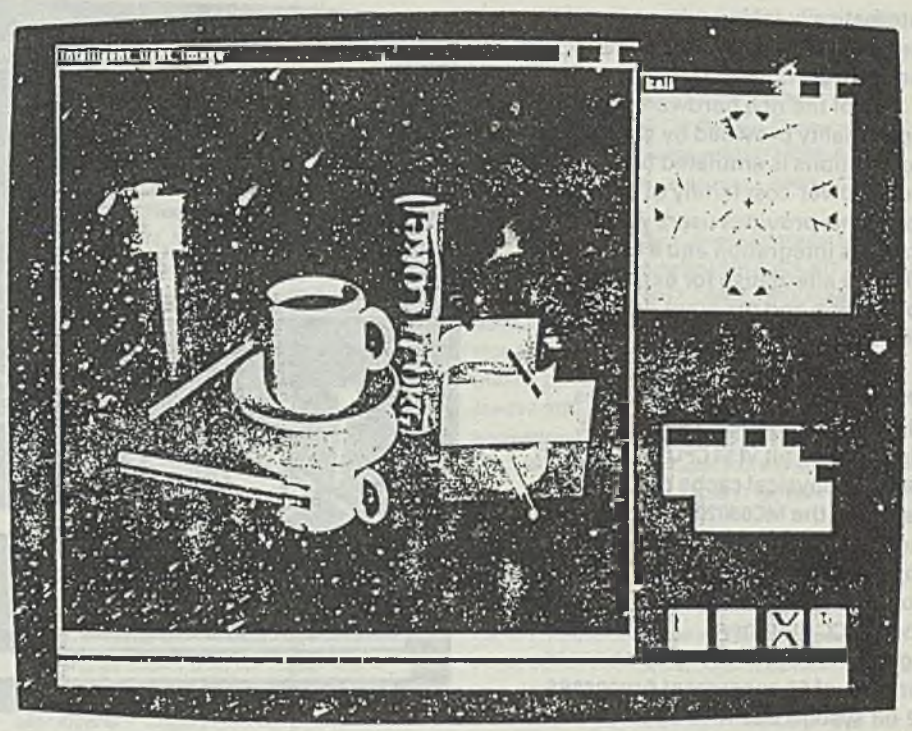
Processor Subsystem
The DN590 TURBO's processor section contains the 32-bit VLSI MC68020 processor that includes a full 32-bit CPU with 16 registers and a 256 byte instruction cache. The MC68020 features 3-stage pipeline processing that allows concurrent execution of up to three instructions. The processor module lets the DN590 support up to 56 concurrent processes with up to 2G bytes of address space per process.

The DN590 TURBO includes the MC68881 floating point coprocessor as standard equipment. The MC68881 performs floating point operations according to the IEEE 754 standard for binary floating point arithmetic. It features an advanced CPU, eight 80-bit registers, three 32-bit control registers, and a coprocessor interface to the MC68020.

The CPU directly addresses memory through an onboard Memory Management Unit (MMU). The MMU is designed to support demand paging requirements from the disk or access to or from the high performance Domain network. This feature allows the Domain network to provide transparent file access around the network, greatly enhancing the sharing of information and graphical data.

The DN590 TURBO is available in 8M and 16M byte ECC memory configurations. Also available is an optional high-performance floating point accelerator (FPX) that transparently increases performance by a factor of up to six depending on operation over the MC68881.

Mass Storage Module
The DN590 TURBO's 5¼ inch 190M byte, dual 190M byte Winchester disks, and ¼ inch 60M byte streaming cartridge tape drive provides local mass storage and convenient, high-speed backup. A single controller supports both the tape and disk units and uses industry standard interfaces to allow the easy integration of larger capacity devices as they become available. The Mass Storage Module uses an 8088 microprocessor, and contains a VME interface and DMA control logic, a battery-backed clock/calendar device, and a separate power supply and fan.



Expansion Module
The DN590 TURBO's option expansion module includes a 4-slot MULTIBUS peripheral adaptor, power supply, and fans. Apollo provides a variety of peripherals for the MULTIBUS adaptor; in addition, users can add devices as appropriate for their applications. Apollo's GPIO software lets application programmers write drivers in high level languages for MULTIBUS devices.

FPX Option
The DN590 TURBO's optional FPX provides a high-performance, industry-standard, IEEE-754/10.0-compliant floating point processor. The FPX uses the Weitek 1164/1165 floating point chip set in a heavily microcoded environment, and performs over 3,800 single-precision Whetstones, and over 2,600 double-precision Whetstones.

In addition, the high performance FPX instruction set can increase performance by as much as 2 to 6 times in floating point intensive applications such as simulation, finite element analysis and modeling.

The FPX option is fully object code-compatible with the MC68881, and transparently runs programs compiled for the MC68881. As a result, recompiling of applications is not required before users and developers can take advantage of the FPX's high performance levels.

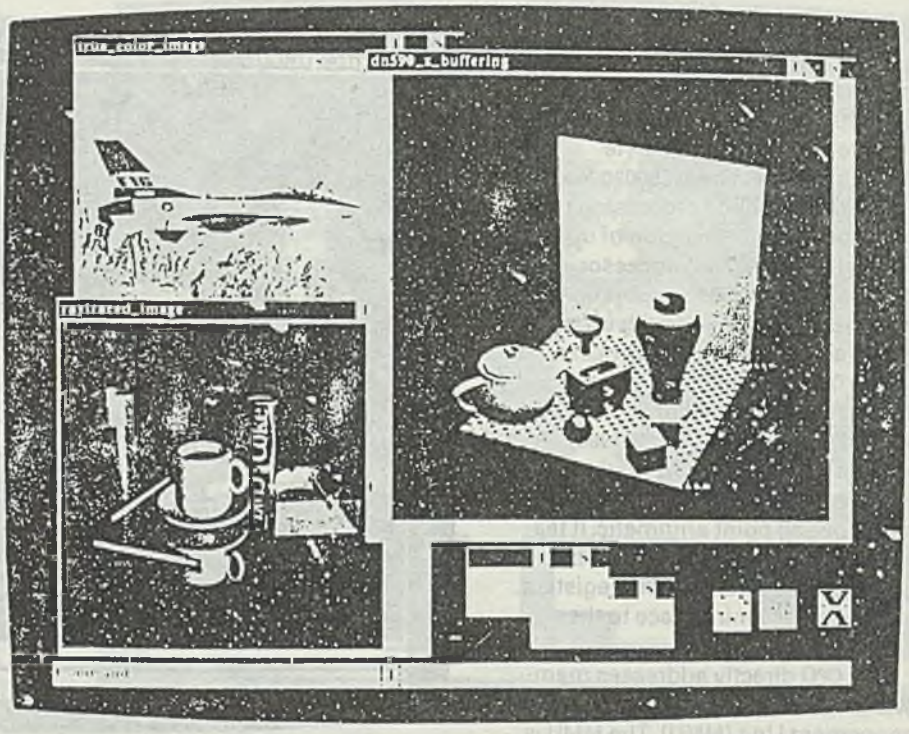
Domain Graphics Resources
Apollo offers application programmers the most comprehensive set of integrated graphics tools in the industry today. The Domain Graphic Resources (DGR™) provide a range of complementary products that form the foundation needed to develop and execute graphics based applications. DGR software can offload routine graphics tasks and simplify the application programming process. In this way, the Domain Graphics Resources speeds the development cycle and optimizes application performance. DGR products address four functional areas; User Environment, Specialized Program Interfaces, Industry Standard Program Interfaces, and Graphics Terminal Emulation.

The Toolkit Solution
Apollo's graphics toolkit reduces the software development effort therefore allowing the solution to be brought to market quickly.

Of course, these sophisticated graphics tools can run on any Apollo workstation without modification,

automatically taking advantage of the specific features and functionality that each system offers.

All of the rich hardware-based functionality provided by the high end workstations is emulated by software on the lower-cost family of workstations. This provides users with total graphics integration and a cost-effective alternative for expensive application and program development efforts.



DN590 TURBO Hardware Features

- High-performance, no-wait state, pipelined 32-bit VLSI CPU design with 16K byte physical cache memory, based on the MC68020 processor and including an MC68881 floating point coprocessor
- Support for up to 16M bytes of high-speed ECC memory
- 2 gigabytes of virtual address space for each of 56 concurrent processes
- 32-bit system bus with up to 14M byte/second bust mode bandwidth
- High-performance FPX floating point accelerator option
- 3DGA graphics accelerator option
- Low-profile, detachable keyboard with programmable keypad and optional mouse
- 155M and dual 155M byte disks for local storage
- Integral 12M bit/second Domain network
- Two RS232C ports (up to 19.2K baud)

Color Display Features

- Double buffering of pseudo color images, with full Z-buffer support
- Advanced lighting model including:
  - Up to 16 light sources
  - 24 plane RBG support
  - Colored light sources
  - Attenuation of light sources
  - Spotlights with attenuation
  - Specularity, highlights
  - Surface properties
- 16.7 million colors displayable, from a palette of 16.7 million colors; 24 color bit-planes planes
- Over 130,000 3-D floating point transformed, clip-checked vectors per second with 3DGA
- Extremely fast, smooth-shaded Z-buffered polygons
- 19 inch tilt and swivel, color display monitor
- High resolution (1280 x 1024 bit map)
- 60Hz non-interlaced refresh rate
- 3DGA renders display lists directly from virtual memory; optional
- Integrated high performance display processor
- Vector generation greater than 16M pixels/second
- Up to 130,000 3-D transformed, floating point, vectors per second with 3DGA
- Smooth shaded Z-buffering and double buffering
- Up to 150,000 2-D integer transformed, clip-checked, vectors per second
- Bit-blts at up to 40M pixels/second
- Sixteen logical raster-ops at up to 30M pixels/second
- Area fills at up to 125M pixels/second
- 2x zoom by pixel replication
- RS343A RGB video output with composite sync on green

DN590 TURBO Peripheral Options

- 5¼ inch 86M byte, 155M, or dual 155M byte Winchester disks
- ¼ inch 60M byte streaming tape cartridge
- Four-slot IEEE 796 MULTIBUS adaptor

The Domain Network

- Baseband network in a ring topology
- Dual address packet with single token arbitration
- Up to 1000 meters between two nodes
- Transparent access to data, programs, and peripherals
- Gateway and bridge access to remote and foreign facilities/protocols

Software

Standard

- AEGIS, Apollo's object-oriented operating system
- Multiple window management with cut and paste
- High-level language debugger
- Domain Graphics software
- Power-on diagnostics

Optional

- Domain/IX, Apollo's twin port of the UNIX standards: Berkeley 4.2 and System V Release 2
- FORTRAN 77, ISO Pascal, Domain/C,™ Domain/CommonLISP™ and Domain/LISP™ support
- Domain Graphics tools
- Database management tools
- Software development tools
- Technical Office Support
- Over 1200 solution-supplier applications packages available

DN590 TURBO Specifications

Physical dimensions:

Electronics Cell: height 62.2 cm (24.5 in.), width 34.3 cm (13.5 in.), depth 72.4 cm (28.5 in.)

Color Monitor: 19-inch, height 53.3 cm (21.0 in.), width 56.4 cm (22.2 in.), depth 54.6 cm (21.5 in.)

Power:

Electronics Cell: *Minimum* – 120 VAC + 10/ – 25%, 7.8A, 50/60Hz +/ – 1Hz, 1,270 watts, 4325 BTU/hr, 220 VAC + 10/ – 25%, 7.7A, 50/60Hz +/ – 1Hz, 1,270 watts, 4325 BTU /hr

Maximum – 220 VAC + 10/ – 25%, 14.2A, 50/60Hz +/ – 1Hz, 1,270 watts, 4325 BTU/hr

Color Monitor: 19-inch – 90 to 132 VAC, 220 watts, 47/63Hz, 750 BTU/hr.

Operating Environment:

Temperature: 15C to 32C (60F to 90F)

Humidity: 20% to 80% relative

Ceiling: 0-2.1 km (0-7000 ft)

North American Sales and Service Offices: *Northeast Region:* Boston (617) 872-4802 & (617) 256-6600, Hartford (203) 287-0155, Newark (201) 535-0900, New York (516) 496-4800, Philadelphia (215) 768-9730, Pittsburgh (412) 471-3905, Rochester (716) 385-6871, Albany (518) 458-7121, *Southeast Region:* Atlanta (404) 393-4720, Austin (512) 328-0230, Baltimore (301) 859-0825, Dallas (214) 869-1491, Houston (713) 871-1991, Orlando (305) 843-8095, Raleigh (919) 544-1715, Tulsa (918) 254-9676, Washington (703) 556-9810, Ft. Lauderdale (305) 946-8011, *North Central Region:* Chicago (312) 882-7030, Cincinnati (513) 831-0590, Detroit (313) 649-8900, Indianapolis (317) 843-5970, Minneapolis (612) 835-4541, St. Louis (314) 569-3332, Wichita (316) 683-3955, *Northwest Region:* Portland (503) 641-6948, Salt Lake City (801) 265-9881, Santa Clara (408) 496-2900, Seattle (206) 453-5544, *Southwest Region:* Denver (303) 792-5700, El Segundo (213) 640-1121, Los Angeles (818) 883-5111, Orange County (714) 768-2988, Phoenix (602) 277-9103, San Diego (619) 458-9921, *Canada: Alberta* – Calgary (403) 234-8737, *Ontario* – Toronto (416) 297-0700, Ottawa (613) 829-1358, *Quebec* – Montreal (514) 739-1781

International Sales and Service Offices: *Australia:* Canberra (61-62) 851 545, Melbourne (61-3) 699 6988, North Sydney (61-2) 957-3382, *Belgium:* Brussels (32-2) 763 0303, *Denmark:* Vedbæk (45-2) 89 09 77, *Finland:* Espoo (358-0) 524 533, *France:* Aix-En-Provence (33-42) 24 40 60, Grenoble (33-76) 495 779, Paris (33-1) 30 58 58 33, Toulouse (33-61) 41 1114, *Germany:* Dusseldorf (49-211) 611 025, Frankfurt (49-69) 66 40 50, Hamburg (49-40) 510 021, Munich (49-89) 92 80 030, Stuttgart (49-711) 71 3045, *Hong Kong:* (852) 5785 785, *Italy:* Milan (39-2) 824 2161, Padova (39-49) 656-144, Rome (39-6) 598-4941, Turin (39-11) 506 906, *Japan:* Osaka (81-6) 251 5491, Tokyo (81-3) 588 1561, Yokohama (81-45) 311 8012, Nagoya (81-52) 204 7361, Sapporo (81-11) 251-1561, *Korea:* Seoul (82-2) 553 4630, *Netherlands:* Houten 31-3403-93211, *Norway:* Oslo (47-2) 23-28-80, *People's Republic of China:* 801-3120, *Singapore:* (65) 345 8633, *Sweden:* Stockholm (46-8) 750 56 20, *Switzerland:* Geneva (41-22) 98 57 88, Zurich (41-1) 833 5060, *Taiwan:* Taipei 886-2776-5590, *United Kingdom: England* – Birmingham (44-21) 550 9010, Richmond (44-1) 948 6055, Manchester (44-61) 962 0222, Milton Keynes (44-908) 366 188, *Scotland* – Livingston (44-506) 416 868

Distributors: *Orion Technologies:* Israel – Givatayim (972-3) 777 719, *Far East Computers (Pte.) Ltd. (FECL):* Singapore (65) 273 8288, *P.T. Canggih Purna Indera:* Indonesia – Jakarta (62-21) 587 171

Corporate Headquarters: Apollo Computer Inc., 330 Billerica Rd., Chelmsford, MA 01824, 617-256-6600, TWX: 710-444-8017, CABLE: APOLLOCO

Canadian Headquarters: Apollo Computer Inc., 1530 Markham Road, Suite 130, Scarborough, Ontario, Canada M1B 3G4, 416-297-0700, FAX: 416-297-1020

International Headquarters: Apollo Computer, S.A., 108, Avenue Louis-Casai, P.O. Box 409, 1215 Geneva, Switzerland (41-22) 98 57 88, TWX: 236 18 ch, FAX: (41-22) 98 58 79

Apollo and Domain are registered trademarks of Apollo Computer Inc. The following are trademarks of Apollo Computer Inc. AEGIS, Domain/IX, Public Domain, Domain/C, Domain/Common LISP, Domain/LISP, DGR, FPX, GPR, 3DGMR, 3DGA.

MULTIBUS is a trademark of Intel Corp. PC is a registered trademark of International Business Machines Corp. UNIX is a registered trademark of AT&T.

The materials contained herein are summary in nature, subject to change, and intended for general information only. Details and specifications concerning the use and operation of Apollo products are available in the applicable technical manuals, available from local sales representatives. Copyright © 1987, Apollo Computer Inc., Chelmsford, MA.

apollo

Janusz Gocałek
Jacek Klauziński
Andrzej Różga

System Programowania Modula-2
dla minikomputera MERA-400
w systemie operacyjnym CROOK-5

1. Charakterystyka systemu Modula-2.

System programowania MODULA-2 składa sie z nastepujacych elementów:

- egzekutora (programu nadzorczego)
- kompilatora jezyka Modula-2
- programu łaczacego (LINKER)
- programów pomocniczych i bibliotek standardowych

Egzekutor jest programem stale rezydujacym w pamieci operacyjnej podczas pracy systemu Modula-2 i zazwyczaj pozostałe programy , zarówno systemowe jak i użytkowe , wykonywane sa pod jego kontrola.

Kompilator umożliwia kompilacje tzw. jednostek kompilacyjnych (modułów), znajdujacych sie w zbiorach zewnetrznych. Jednostka kompilacyjna jest źródłowa postać modułu :

- definicyjnego
- implementacyjnego
- programowego.

Skompilowany program słowny (jest nim najcześciej moduł programowy) przed załadowaniem do pamieci operacyjnej musi zostać połaczony ze wszystkimi , oddzielnie skompilowanymi modułami , z których importuje on obiekty. Połaczenie skompilowanych modułów oraz ostateczne formatowanie postaci programu dokonuje sie podczas pracy programu łaczacego.

Programy znajdujace sie w zbiorach wygenerowanych przez program łaczacy (LINKER) mosa być załadowane do pamieci operacyjnej i uruchomione przez egzekutor systemu. Programy w tej postaci mosa być także załadowane do pamieci operacyjnej i uruchomione przez inne, aktualnie wykonywane , programy użytkowe jako ich nakładki (overlays).

Ostatnim , lecz niezwykle istotnym , składnikiem systemu jest zbiór plików zawierajacych moduły biblioteki systemu. Moduły znajdujace sie w tych zbiorach , udostepniajace miedzy innymi operacje we/wy, mosa być importowane przez programy użytkowe. Program łaczacy automatycznie dołacza odpowiednie moduły biblioteczne do modułu programowego.

W systemie programowania Modula-2 przyjeto pewien zbiór standardowych rozszerzeń nazw zbiorów.

Rozszerzenia zbiorów jakie przyjeto we wzorcowej wersji jezyka Modula-2, w tej realizacji zostały zmienione. Różnica wynika z własności systemu operacyjnego. Przyjeto rozszerzenia zbiorów

dwuliterowe poprzedzone znakiem "%" (%LN, %LD itd.). Każde z tych rozszerzeń określa kategorię zawartości zbioru informacji znajdujących się w zbiorze, np. rozszerzenie %DF dla zbiorów zawierających źródłową postać modułów definicyjnych, czy rozszerzenie %LD dla zbiorów zawierających skompilowane i połączone programy.

Stosowanie standardowych rozszerzeń jest wygodne, pozwala bowiem na łatwe określenie rodzaju zawartości zbioru, a ponadto pozwala na pomijanie rozszerzeń podczas specyfikacji plików wejściowych dla programów systemowych. Przyjęta w systemie konwencja dotycząca rozszerzeń nazw zbiorów nie jest obligatoryjna i użytkownik ma pełną swobodę w wybieraniu rozszerzeń nazw wykorzystywanych zbiorów.

Określenie zbiorów wejściowych dla uaktywnionych programów systemowych odbywa się według takiego samego, dla wszystkich programów , schematu. Na monitorze ekranowym wyświetlane zostaje pytanie o nazwę zbioru zawierającego określone dane. Wprowadzając z klawiatury nazwę odpowiedniego zbioru, w przypadku niektórych programów użytkownik może także dołączyć nowe opcje, określające tryb pracy programu. Opcje wprowadza się bezpośrednio po nazwie programu, poprzedzając każdą z nich znakiem "?". Opis znaczenia opcji poszczególnych programów zamieszczono w dalszej części opracowania.

Fakt zakończenia wprowadzania nazwy zbioru , użytkownik sygnalizuje wysłaniem znaku (cr). Procedura odczytująca nazwę zbioru , w każdym przypadku zna standardowe rozszerzenie jego nazwy oraz w niektórych przypadkach także standardową nazwę właściwą. Jeżeli użytkownik , wprowadzając nazwę zbioru , pominie jeden z jej składników, automatycznie przyjmowany jest odpowiedni składnik nazwy standardowej. W niektórych przypadkach , np. podczas kompilacji, przy wprowadzaniu nazw zbiorów zawierających importowane moduły, możliwe jest nawet całkowite pominięcie nazwy zbioru i wysłanie jedynie znaku sterującego (cr) (kompilacja z opcją Query).

W takich przypadkach nazwa właściwa tworzona jest z pierwszych sześciu liter nazwy poszukiwanego modułu (dlatego celowe jest nadawanie takich samych nazw modułom i zbiorom , które je zawierają).

Kompilator i program łączący mogą poszukiwać niezbędnych im zbiorów (kompilator - zbiorów symboli, program łączący - zbiorów z kodem oddzielnie skompilowanych modułów) według dwóch strategii:

Strategia standardowa polega na automatycznym poszukiwaniu potrzebnych zbiorów wg. nazw standardowych. Podczas takiego poszukiwania wyświetlana jest informacja o jego rezultacie, jednak w przypadku niepowodzenia użytkownik nie ma możliwości wprowadzenia nowej nazwy.

Druga strategia , wymuszona opcją Q , polega na wprowadzeniu odpowiednich nazw zbiorów przez użytkownika. Wysyłając znak cr, użytkownik może spowodować poszukiwanie zbioru wg. nazwy standardowej, jednak w przypadku niepowodzenia ma on możliwość powtórnego wprowadzania nazwy tego samego zbioru.

Zbiory wyjściowe programów systemowych posiadają tę samą nazwę właściwą co zasadniczy zbiór wejściowy. Zbiory te otrzymują standardowe, dla danej kategorii danych , rozszerzenie nazwy.

2. MODULA - Program nadzorczy systemu.

Egzekutor systemu programowania Modula-2 jest jedynym programem tego systemu, ładowanym do pamięci operacyjnej przez system operacyjny CROOK-5.

Jest on programem stale rezydującym w pamięci operacyjnej podczas pracy systemu. Udostępnia on własny język komend, dzięki któremu użytkownik może żądać załadowania i wykonania odpowiednich programów (systemowych lub użytkowych).

Na ogół wszystkie programy wykonywane są pod kontrolą egzekutora. Jego głównym zadaniem jest udostępnienie wykonywanym programom odpowiedniej maszyny wirtualnej. Maszyna ta posiada pełną listę rozkazów minikomputera Mera-400. Maszyna wirtualna udostępnia ponadto inne wirtualne rozkazy, które między innymi wspomagają realizację operacji eksportowanych przez standardowy moduł SYSTEM (np. TRANSFER, NEWPROCES, ADR itd.).

Do innych zadań egzekutora należy ładowanie programów do pamięci operacyjnej, inicjowanie ich wykonania oraz reagowanie na ewentualne błędy, jakie mogły zaistnieć podczas wykonywania programów.

Egzekutor sygnalizuje gotowość do pracy programu, wyświetlając na monitorze znaki "=>". Nazwa zbioru zawierającego połączony i skompilowany moduł programowy jest jedynym typem komendy akceptowanym przez egzekutor. Wyszczególnienie nazwy zbioru oznacza żądanie załadowania i uruchomienie programu znajdującego się w tym zbiorze. Wprowadzając nazwę zbioru można pominąć jej rozszerzenie . Brakujące elementy nazwy uzupełniane są w sposób standardowy. Po zakończeniu wykonywania uaktywnionego programu , egzekutor jest gotowy do przyjęcia następnej komendy.

W przypadku wystąpienia błędu w trakcie wykonywania programu , egzekutor kopiuje obraz pamięci operacyjnej do zbioru (w celu późniejszej analizy sytuacji przez program diagnostyczny) oraz wyświetla komunikat o przyczynie zakończenia wykonywania programu.

Skompilowany i połączony program może zostać załadowany nie tylko przez egzekutor systemu, ale także przez inny, wykonywany program użytkowy. Program taki stanowi więc nakładkę dla programu, przez który został załadowany i uaktywniony. Programy-nakładki mogą importować obiekty z programów , które rezydują w pamięci.

Nakładki są typowymi modułami programowymi (lub implementacyjnymi). Skompilowany moduł programowy może więc być zarówno programem głównym , ładowanym przez egzekutor , jak i nakładką innego programu użytkowego. Decyzja o tym , czy dany moduł będzie ładowany bezpośrednio przez egzekutor (jako program główny), czy też przez inny program użytkowy (jako jego nakładka), podejmowana jest dopiero na etapie łączenia tego modułu.

Programy użytkowe ładujące własne nakładki wykorzystują zwykle do tego celu operacje zdefiniowane w module Program (moduł napisany w języku Modula-2). Program uaktywniający nakładkę może sam obsługiwać ewentualne sytuacje wyjątkowe, jakie mogły zaistnieć podczas wykonywania tej nakładki lub może zlecić obsługę błędu programowi nadzorczemu. Każdy z wykonywanych programów może, korzystając z modułu program , ładować własne nakładki niezależnie od tego , czy stanowi on program główny, czy też sam jest nakładką innego programu. W pamięci operacyjnej może więc w danej chwili znajdować się cała hierarchia załadowanych i uaktywnionych nakładek, ograniczona jedynie fizycznym rozmiarem dostępnej pamięci.

Struktura egzekutora.

Egzekutor systemu Modula-2 jest programem, który rezyduje na stałe w pamięci operacyjnej podczas pracy systemu. Pozostałe programy systemowe lub użytkowe , wykonywane są pod kontrolą programu nadzorczego, są jego nakładkami. Każdorazowo po zakończeniu wykonywania programu, użytkowego lub systemowego, egzekutor ładuje do pamięci operacyjnej własną nakładkę , która stanowi program interpreter komend użytkownika (chyba że program, który zakończył pracę zażądał następnika).

Egzekutor systemu złożony jest z dwóch części. Jedna z nich stanowi program wspomagający wykonywanie programów wygenerowanych przez kompilator języka Modula-2. Program ten przygotowano w języku ASSEMBLER-CROOK. Druga częścia egzekutora jest tzw. Program-Monitor , zaprogramowany w języku Modula-2 , który steruje pracą całego systemu. Obie części egzekutora komunikują się przez wspólnie wykorzystywany obszar pamięci, który jest zdefiniowany przez odpowiedni moduł o nazwie CROOK.

Program wspomagający.

Zadaniem programu wspomagającego jest realizacja , na rzecz pozostałych programów , odpowiedniej maszyny wirtualnej. Maszyna ta , oprócz rozkazów wykonywanych przez procesor fizyczny , udostępnia szereg innych rozkazów , które są generowane przez kompilator języka Modula-2. Procesy mogą żądać realizacji rozkazu wirtualnego, wykonując rozkaz przerwania programowego. Argument rozkazu (liczba) określa jeden z rozkazów wirtualnych. Do rozkazów wirtualnych należą między innymi rozkazy:

Testowania wyliczonych wartości indeksów i testowania przepełnienia stosu (generowane przez kompilator po ustawieniu odpowiednich przełączników w programie źródłowym), rozkazy kopiowania: łańcuchów danych , parametrów dynamicznych oraz parametrów procedur przekazywanych przez wartość; a także rozkazy realizujące standardowe procedury języka , np. TRUNC, FLOAT, HALT.

Program wspomagający zawiera jądro systemu zarządzania procesami. W jądrze tym zdefiniowano procedury manipulowania korutynami: NEWPROCES, TRANSFER. Procedury te uaktywniane są za pomocą rozkazów maszynowych , generujących przerwania programowe.

Program wspomagający jest nie tylko składnikiem egzekutora systemu, ale także składnikiem wszystkich programów wykonywanych bez nadzoru egzekutora.

Monitor systemu.

Zadaniem monitora jest kierowanie kolejnością wykonywania programów, ładowanie programów ze zbiorów zewnętrznych do pamięci operacyjnej i ich uaktywnianie oraz obsługa ewentualnych sytuacji, jakie mogą zaistnieć podczas wykonywania programów.

Do pamięci operacyjnej , na przemian z programami użytkowymi (i systemowymi) ładowany jest interpreter komend. Jego zadaniem jest wyświetlenie na monitorze informacji o rezultacie wykonania ostatnio aktywnego programu oraz przeprowadzenie z operatorem konwersacji, której celem jest określenie zbioru, z którego załadowany zostanie następny program do wykonania.

3. COMP - Kompilator Jezyka Modula-2.

Kompilator Jezyka Modula-2 , wchodzacy w skład systemu programowania , można uaktywnić zleceniem "COMP" programu nadzorczego. Po uruchomieniu kompilatora , zgłasza sie on komunikatem "source file) ", oczekujac na wprowadzenie nazwy zbioru zawierajacego jednostke kompilacyjna (źródłowa postać modułu definicyjnego , implementacyjnego lub programowego). Wprowadzajac nazwe zbioru, można dodatkowo wyszczególnić pewna liczbe opcji , określajacych tryb pracy kompilatora.

Podczas kompilacji modułu definicyjnego kompilator generuje tzw. zbiór symboli, zawierajacy informacje o identyfikatorach zadeklarowanych w kompilowanym module oraz zbiór tekstowy, zawierajacy ponumerowane wiersze źródłowej postaci modułu. Zbiór symboli wykorzystywany jest podczas kompilacji innych modułów , importujacych obiekty z danego modułu definicyjnego, w celu sprawdzenia zgodności odwołań i kompatybilności typów. Zbiór tekstowy przeznaczony jest dla użytkownika.

Podczas kompilacji modułu implementacyjnego (lub programowego) generowane sa trzy nowe zbiory. Pierwszy zawiera przekład kompilowanego modułu na jezyk maszynowy. Drugi, zbiór referencji, generowany przez kompilator zawiera informacje o nazwach obiektów (modułów, procedur, zmiennych , typów) występujacych w module. Zbiór ten wykorzystywany jest jedynie przez program diagnostyczny (debugger). Trzeci, zbiór tekstow, przeznaczony jest dla użytkownika i oprócz ponumerowanych wierszy źródłowej postaci modułu , zawiera także wzgledne adresy rozkazów maszynowych, na które przetłumaczono kolejne instrukcje modułu. Zbiór tekstowy może być również wykorzystywany przez program diagnostyczny, w celu wyświetlenia użytkownikowi odpowiednich fragmentów źródłowej postaci modułu, podczas analizy obrazu pamieci operacyjnej, składowanego po awarii wykonywanego programu.

Podczas kompilacji kontrolowane sa zbiory symboli wszystkich modułów, które sa importowane przez moduł podlegajacy kompilacji , a w przypadku kompilacji modułu implementacyjnego, także zbiór symboli jego własnego modułu definicyjnego.

Ewentualne błędy w module źródłowym mogą zostać wykryte przez kompilator podczas jednego z trzech pierwszych przebiegów jego pracy. W tym przypadku kompilacja zostaje zakończona po trzecim przebiegu , a w wypadku włączenia opcji ?l generowany jest zbiór tekstowy, zawierajacy informacje o poszczególnych błędach i miejscach w których je wykryto.

Standardowo wyświetlane sa na monitorze błędne wiersze programu, wraz z sygnalizacja błędów.

4. LINK - Program łączacy.

Program łączacy wykorzystywany jest do łączenia skompilowanego modułu stanowiacego program główny lub implementacyjny ze wszystkimi , oddzielnie skompilowanymi, modułami, które sa importowane przez program. Moduł zawierajacy program główny musi ponadto zostać połączony z baza . Baze stanowi program, który bedzie ładował do pamieci operacyjnej i uaktywniał łączony program. Dla wiekszości programów (dla programów , które nie sa nakładkami innych programów użytkowych) baza jest program nadzorczy systemu. Zbiór w którym znajduje sie skompilowany i połączony program stanowiacy baze dla innego programu, nazywany jest zbiorem bazowym tego programu (BASE FILE).

Program łączący można załadować do pamięci operacyjnej i uaktywnić zleceniem "LINK" egzekutora systemu. Po uruchomieniu program łączący wyświetla na monitorze napis "master file)" i oczekuje na wprowadzenie nazwy zbioru zawierającego skompilowany moduł programu głównego. Do wprowadzonej nazwy zbioru można dołączyć opcje, określające tryb pracy programu łączącego.

Program łączący generuje zbiór zawierający, odpowiednio sformatowany i uzupełniony, moduł programu głównego i inne importowane przez ten program moduły, które zostały skompilowane osobno (chyba że moduły te zostały już wcześniej przyłączone do niższych warstw w hierarchii nakładek, np. do programu-bazy). Program znajdujący się w tym zbiorze jest gotowy do załadowania do pamięci operacyjnej i uruchomienia przez program bazę.

Podczas łączenia modułów, program łączący może wygenerować zbiór z tzw. mapą pamięci, przeznaczoną dla użytkownika. Mapa pamięci zawiera informacje o adresach pamięci operacyjnej, jakie przydzielono danym i procedurom poszczególnych modułów.

Program łączący musi mieć dostęp do zbioru bazowego łączonego programu, w celu ustalenia obszaru pamięci operacyjnej, do którego będzie w przyszłości ładowany łączony program oraz w celu sprawdzenia, czy importowane przez program moduły nie zostały już połączone z bazą (tzn.czy będą one znajdować się w pamięci operacyjnej podczas ładowania programu). Zbiorem bazowym dla programów ładowanych bezpośrednio przez egzekutor systemu jest zbiór zawierający kod fragmentu programu nadzorczego (plik ze skompilowanym i połączonym monitorem systemu). Zbiór ten jest znany programowi łączącemu i jest automatycznie wykorzystywany jako zbiór bazowy podczas standardowego przebiegu łączenia.

Jeżeli łączony program ma stanowić nakładkę dla innego programu użytkowego (programy ładowane i uaktywniane przez program nadzorczy są w rzeczywistości nakładkami egzekutora), program łączący powinien zostać uruchomiony z odpowiednią opcją. W tym przypadku program łączący pyta użytkownika o nazwę zbioru bazowego dla łączonego programu.

Jeżeli łączony program importuje inne, oddzielnie kompilowane moduły, a nie zostały one wcześniej przyłączone do programu-bazy, to program łączący może automatycznie wyszukiwać potrzebne zbiory z dysku lub żądać od użytkownika podania określonych nazw zbiorów, które zawierają te moduły.

5. Biblioteka systemu.

Język programowania modula-2 jest językiem wysokiego poziomu, pozwalającym na projektowanie zarówno oprogramowania użytkowego, jak i systemowego. Ponieważ modula-2 jest językiem maszynowo niezależnym, operacje charakterystyczne dla różnych typów komputerów nie mogą być elementem samego języka. Korzystając z tzw. udogodnień niskiego poziomu języka modula-2, można jednak zaprogramować odpowiednie operacje wysokiego poziomu, pozwalające na przeprowadzenie transmisji we/wy, konwersję postaci danych, dynamiczną alokację pamięci operacyjnej, manipulowanie zbiorami, szeregowanie procesów itp. Operacje maszynowo-zależne, dzięki koncepcji modułów wprowadzonej do języka, mogą być ograniczone i izolowane. Z uwagi na fakt, że wspomniane wyżej operacje są komponentami większości programów, biblioteka standardowa modułów jest istotną częścią implementacji systemu programowania modula-2.

- moduły systemowe (udostępniające powiązanie języka z systemem operacyjnym i maszyną)

- moduły realizujące operacje we/wy
- moduły realizujące operacje zmiennoprzecinkowe
- moduły realizujące działania na tekstach
- moduły wspomagające wieloprogramowanie

Biblioteka systemu programowania modula-2 została podzielona na kilka poziomów hierarchii całej biblioteki. Najniższym poziomem w tej hierarchii są moduły realizujące tzw. udogodnienia niskiego poziomu języka. Do nich zaliczamy następujące moduły:

FileSystem
moduł realizujący sprzęg języka z systemem operacyjnym i maszyną

Terminal
moduł realizujący obsługę terminala i operacje z nim związane

Program
moduł realizujący ładowanie programów oraz nakładek do pamięci operacyjnej

Storage
moduł definiujący operacje dynamicznego przydziału i zwalniania pamięci

Clock
moduł realizujący odczyt czasu i daty z zegara maszyny i datownika systemowego.

Exceptions
moduł obsługujący sytuacje wyjątkowe.

Język modula-2 nie posiada żadnych inststrukcji we/wy , jako instrukcji wbudowanych w język. To skrajne podejście możliwe jest z dwóch powodów. Po pierwsze istnieje struktura modułowa , pozwalająca na skonstruowanie hierarchii modułów reprezentujących różne poziomy abstrakcji.

Po drugie , modula-2 dopuszcza wyrażanie operacji specyficznych dla danego komputera, takich jak komunikacja z urządzeniami zewnętrznymi. Operacje takie umieszczane są zwykle na najniższym poziomie hierarchii i są zaliczane do tzw. udogodnień niskiego poziomu języka. Program pragnący ignorować szczegóły obsługi urządzeń , importuje procedury komunikacji z wyższych poziomów hierarchii, natomiast program wykorzystujący specyficzne cechy danego urządzenia, importuje procedury z modułów niższego poziomu (aż do operowania na fizycznych rejestrach systemu komputerowego włącznie).

W omawianym systemie programowania , moduły biblioteczne dostarczają procedury realizujące sekwencyjne operowanie na danych. W modułach z wyższych poziomów hierarchii , zespoły danych są strukturami , które nie istnieją jako podstawowe struktury danych w języku modula-2. Struktury takie są zdefiniowane w module filesystem.

Funkcje realizowane przez przedstawione dalej moduły wejścia-wyjścia można podzielić na trzy klasy. Jedna z nich stanowi faktyczna transmisja danych, wymagająca aktywacji niektórych urządzeń zewnętrznych. Druga grupę stanowią funkcje związane z transformacją reprezentacji danych, np. wartość typu cardinal, reprezentowana w komputerze w postaci binarnej przez jedno słowo maszynowe, przed wyświetleniem na monitorze ekranowym musi zostać przekształcona na sekwencję bajtów, będących kodami znaków

cyfr dziesietnych (np. w kodzie ascii). Transformacja taka jest niezależna od stosowanych urzadzeń zewnetrznych i dlatego powinna być odseparowana od operacji specyficznych dla danych urzadzeń zewnetrznych. Do trzeciej klasy należa funkcje zwiazane z urzadzeniami, z którymi może być stowarzyszona jednocześnie wieksza liczba strumieni. W przypadku pamieci dyskowej, operacje tej grupy dotycza przydziału obszarów pamieci dla zbiorów , nazywania zbiorów , zarzadzania kartoteka zbiorów. Przydział pamieci zewnetrznej i zarzadzanie kartoteka zbiorw sa zwykle zadaniami systemu operacyjnego. Z uwagi na różnorodność rozwiazań tych operacji w różnych systemach , postulowanie obowiazujacych konwencji dla podstwowych sposobów manipulowania zbiorami , niezależnych od systemu operacyjnego , a jednocześnie łatwych do zaimplementowania w każdym systemie , jest rzecza szczególnie trudna. Problem ten rozwiazano , oferujac użytkownikowi hierarchie modułów i zostawiajac programiście wybór poziomu wejścia do tej hierarchii. Wejście na wysokim poziomie daje zalete prostoty pojeć i przenośności oprogramowania, natomiast wejście na niskim poziomie udostepnia pełen zakres możliwości oferowanych przez zainstalowany system operacyjny.

Literatura.

1. Niklaus Wirth
Programming in Modula-2.
New York, Springer-Verlag 1983.

2. E.Knepley , R.Platt
Modula-2 Programming.
Reston Publishing, Company Inc.
A Prentice Hall Company.
Reston, Virginia USA 1985.

3. R.Wiener, R.Sincovec
Software Engineering with Modula-2 and Ada.
University of Colorado at Colorado Springs.
John Wiley and Sons New York, 1984.

4. Kaare Christian
A Guide to Modula-2.
Springer-Verlag 1986.

5. Daniel Thalmann
Modula-2. An Introduction.
Springer-Verlag 1985.

6. J.Gocałek, J.Klauziński
MODULA-2: Miedzy Pascalem a Ada.
II Konferencja Użytkowników Minikomputera
Mera-400. Materiały, 25-27.10.1984

Janusz Gocałek
Jacek Klauziński
Andrzej Rózga

Różnice pomiedzy jezykami Modula-2 i Pascal.

1. Słownik jezyka.

- wszystkie znaki identyfikatorów sa znaczace,
- rodzaj liter (małe/duże) identyfikatora jest znaczacy, w szczególności słowa kluczowe sa pisane dużymi literami,
- znaki specjalne (np. podkreślenie) nie mogą być użyte w identyfikatorach,
- liczby mogą być przedstawione w postaci dziesietnej (49), ósemkowej (33B) lub szesnastkowej (1DH); pierwszym znakiem stałej szesnastkowej musi być cyfra. Stałe znakowe (typu CHAR) mogą być zapisywane w postaci wartości ósemkowej z przyrostkiem "C" (tzn. 'A' = 101C),
- napisy (tzw. ciągi znaków ujete w apostrofy lub cudzysłów) sa traktowane jako tablice znaków indeksowane liczbami całkowitymi (tzn. jako stałe typu ARRAY [0..N-1] OF CHAR). Napis nie może zawierać jednocześnie apostrofów i cudzysłowów. Napis musi zawierać przynajmniej dwa znaki; pojedyńczy znak w cudzysłowiu (lub ujety w apostrofy) traktowany jest jak stała typu CHAR.
- nowymi słowami kluczowymi w Moduli-2 są:

BY	FROM	MODULE
DEFINITION	IMPLEMENTATION	POINTER
ELSIF	IMPORT	QUALIFIED
EXIT	LOOP	RETURN
EXPORT		

- następujace słowa nie sa wykorzystywane jako słowa zastrzezone w jezyku Modula-2:

DOWNTO	LABEL
FILE	PACKED
FUNCTION	PROGRAM
GOTO	TEXT

- NIL bedac słowem kluczowym w Pascalu jest identyfikatorem standardowym w jezyku Modula-2,
- symbol # jest synonimem symbolu <> (różny),
- symbol & jest synonimem symbolu AND,
- komentarze sa ograniczone symbolami (* *), komentarze mogą być zagnieżdżone,

2. Deklaracje i zakres identyfikatorów.

- zakłada sie , że identyfikatory standardowe sa zadeklarowane w wyimaginowanej procedurze otaczajacej program. Z tego wzgledu identyfikatory te nie mogą być redefiniowane w cie-

le modułu , lecz mogą być redefiniowane wewnątrz procedury,
- moduły mogą exportować identyfikatory, których zakres zostaje wtedy rozszerzony na cały moduł importujący te identyfikatory,
- importowane identyfikatory mogą być prefiksowane nazwą modułu , z którego zostały importowane. Dzięki temu możliwe jest importowanie identycznych identyfikatorów z różnych modułów,
- nowymi identyfikatorami w języku Modula-2 są:

BITSET	DEC	HALT	INC	PROC
CARDINAL	EXCL	HIGH	INCL	VAL
CAP	FLOAT			

- bloki deklaracji identyfikatorów mogą występować w dowolnej kolejności , pod warunkiem że identyfikatory wykorzystywane w definicji zostały uprzednio zadeklarowane. Pozwala to np. na grupowanie deklaracji typów i odpowiadających im zmiennych,

3. Stałe.

- wyrażenia złożone ze stałych mogą występować wszędzie tam , gdzie dopuszczalne jest wystąpienie stałej. Wyrażenia takie nie mogą zawierać zmiennych lub wywołań funkcji,
- wyrażenia złożone ze stałych mogą zawierać stałe typu zbiorowego (elementy zbioru ujęte są w nawiasy sześcienne,
- stałe typu zbiorowego musi poprzedzać identyfikator typu zbiorowego (z wyjątkiem zbioru standardowego BITSET),
- elementami stałych typu zbiorowego mogą być jedynie stałe lub wyrażenia złożone ze stałych.

4. Deklaracje typu.

- nowymi typami standardowymi są: CARDINAL i BITSET,
- dolne i górne ograniczenie typu okrojonego ujęte są w nawiasy kwadratowe (np. [1..10]),
- typy indeksów tablic nie są ujmowane w nawiasy (np. ARRAY Tab OF ListNode),
- typ CHAR obejmuje zbiór ASCII,
- rekordy mogą zawierać wiele części zmiennych. Części stałe i części zmienne rekordu mogą być definiowane w dowolnej kolejności. Listy pól wariantów nie są ujmowane w nawiasy. Separatorem wariantów jest symbol "|". Pola zmienne zakończone są słowem kluczowym END,
- w deklaracjach typów wskaźnikowych zamiast symbolu "" wykorzystywane są słowa kluczowe POINTER TO,
- deklaracja typu wskaźnikowego nie musi zawierać identyfikatora typu zmiennej wskazywanej przez wskaźnik. W miejscu tego identyfikatora może występować definicja typu zmiennej wskazywanej,
- procedury (funkcje) są typami. Możliwa jest deklaracja zmiennej typu procedura (funkcja). Standardowy identyfikator PROC jest identyfikatorem typu bezparametrowej procedury,
- nie istnieją standardowe typy FILE i TEXT,

5. Wyrażenia.

- wprowadzono nowy operator teoriomnogościowy "/" oznaczający różnicę symetryczną,
- pary operatorów: & i AND oraz ≠ i <> są synonimami,
- jeżeli wartości pierwszych operandów wyrażenia logicznego implikują wartość całego wyrażenia , to pozostałe operandy nie są wartościowane. Konsekwencją tego jest fakt, że kolejność operandów wyrażenia logicznego może być znaczącą (jeżeli np. wartości niektórych operandów są nieokreślone),
- operandy wyrażenia muszą być zgodnych typów (wyjątkiem są wyrażenia z operandami typu CARDINAL i ADDRESS oraz POINTER i ADDRESS),
- funkcje konwersji typów umożliwiają stosowanie operandów różnych typów w tym samym wyrażeniu,
- operatory relacyjne określone są na operandach typu podstawowego: INTEGER, CARDINAL, BOOLEAN, CHAR, REAL, oraz
- na operandach typu wyliczeniowego i okrojonego. Operatory "=" i "≠" określone są dodatkowo na operandach typu zbiorowego i wskaźnikowego a operatory "<=" i ">=" na operandach typu zbiorowego,

6. Instrukcje.

- wszystkie instrukcje strukturalne zakończone są słowem kluczowym. Słowem tym jest END, z wyjątkiem instrukcji REPEAT, która kończy się klauzulą UNTIL,
- wprowadzono słowo kluczowe ELSIF aby uniknąć wielokrotnego powtarzania słowa kluczowego END na końcu instrukcji warunkowej z wieloma warunkami,
- elementy listy wyboru instrukcji CASE oddzielone są od siebie symbolem "|". Etykiety wyboru mogą być zapisane w postaci okrojonego zbioru wartości (np. "a".."b"). W instrukcji CASE wprowadzono opcjonalną klauzulę ELSE,
- instrukcja FOR posiada opcjonalną klauzulę BY pozwalającą na zwiększenie lub zmniejszenie zmiennej sterującej o wyszczególnioną stałą. Klauzula DOWNTO nie jest wykorzystywana. Zmienne sterujące mogą być typu wyliczeniowego,
- instrukcja WITH nie może zawierać listy zmiennych, instrukcje WITH mogą być zagnieżdżone explicite,
- wprowadzono instrukcję RETURN kończącą wykonanie funkcji (lub procedury) i instrukcję EXIT kończącą wykonywanie instrukcji pętli LOOP,
- nie istnieje instrukcja GOTO,
- w instrukcji podstawienia można jednocześnie stosować operandy typu CARDINAL i INTEGER,

7. Procedury.

- parametr formalny procedury może być typu ARRAY OF (type). Zakres indeksu tablicy będącej parametrem aktualnym [a..b] zostanie odwzorowany na zakres [0..b-a] tablicy będącej parametrem formalnym. Górne ograniczenie (b-a) można wyznaczyć wykorzystując funkcję standardową HIGH,
- w języku nie występuje słowo kluczowe FUNCTION. Funkcja jest procedurą w deklaracji której zdefiniowano również typ wyniku. Wywołanie funkcji musi zawierać listę parametrów aktualnych (być może pusta),
- w języku nie istnieją standardowe procedury wejścia/wyjścia Niezbędne procedury i funkcje mogą być importowane z modułów bibliotecznych,
- wprowadzono nowe procedury standardowe:

CAP	HALT	INCL
DEC	HIGH	VAL
EXCL	INC	

8. Moduły.

- w języku Modula-2 wprowadzono pojęcie modułu. Moduły mogą zawierać deklaracje obiektów: stałych, typów, zmiennych, procedur oraz ciągu instrukcji. Moduły pozwalają na ukrycie szczegółów implementacji zdefiniowanych w nich operacji, ułatwiając tym samym projektowanie i analizę złożonych programów,
- do języka wprowadzono standardowy moduł SYSTEM, który eksportuje typy: WORD, ADDRESS, PROCESS i procedury: ADR, SIZE, TSIZE, NEWPROCESS, TRANSFER,
- w realizacji Systemu Programowania Modula-2 dla MERY-400 moduł SYSTEM eksportuje dodatkowo procedurę CROOKCALL.

9. Przykład.

```
MODULE Test; (* Modula-2)      | program test; { Pascal }
FROM InOut IMPORT              |
           WriteLn;            |
           WriteInt,           |
           WriteString;        |
CONST                          | const
  Size = 8190;                 |   Size = 8190;
VAR                            | var
  Flags : ARRAY [0..Size]      |   Flags : array [0..Size] of
          OF BOOLEAN;          |           boolean;
  i,K,Prime,                   |   i,K,Prime,
  Count,Iter : INTEGER;        |   Count,Iter : integer;
BEGIN                          | begin
  WriteLn;                     |
  WriteString('10 iter');      |   writeln('10 iter');
  FOR Iter:=1 TO 10 DO         |   for Iter:=1 to 10 do
                               |    begin
  Count:=0;                    |     Count:=0;
  FOR I:=0 TO Size DO          |     for I:=0 to Size do
                               |      begin
   Flags[I]:=TRUE              |       Flags[I]:=True;
  END;                         |      end;
  FOR I:=0 TO Size DO          |     for I:=0 to Size do
                               |      begin
   IF Flags[I] THEN            |       if Flags[I] then
                               |        begin
    Prime:=I+I+3;              |         Prime:=I+I+3;
    K:=I+Prime;                |         K:=I+Prime;
    WHILE K<=Size DO           |         while K<=Size do
                               |          begin
     Flags[K]:=FALSE;          |           Flags[K]:=False;
     K:=K+Prime;               |           K:=K+Prime;
    END;                       |          end;
    Count:=Count+1;            |         Count:=Count+1;
   END;  (* IF *)              |        end; { if }
  END;  (* FOR I *)            |      end; {for I }
 END;  (* FOR Iter *)          |     end; { for iter }
 WriteLn;                      |
 WriteInt(Count,6);            |
 WriteString(' primes');       |   writeln(Count,' primes')
END Test.                      |  end. {test}
```

Literatura

[1]. David W.Carroll
Modula-2 versus Pascal for Microcomputers:
An Update.
Software Tools Vol.10
Dr Dobb's Journal, November 1985, Canada.

[2]. H.McLarty, David W.Smith
An Introduction to Modula-2 for Pascal
Programmers.
Software Tools Vol.9
Dr Dobb's Journal, May 1984 , Canada.

[3]. David Coar
Pascal , Ada and Modula-2.
Byte, August 1984, pp. 215-232.

[4]. David V. Moffat.
UCSD Pascal Vs. Modula-2 a dissenting view.
Byte, August 1984, pp. 428-430.

[5]. Smedema, Medema, Boasson
Programming Language Pascal, Modula, Chill and Ada.
Prentice-Hall 1984.

[6]. Richard Gleaves
Modula-2 for Pascal Programming.
Springer-Verlag Berlin, 1985.

Literatur

[1]. David W.Carroll
Modula-2 versus Pascal for Microcomputers.
In: Byte 6, Januar 1981, Seite 160 ff.

[2]. Niklaus Wirth
Modula-2 und Lilith.
Springer-Verlag, 1982.

[3]. Dr. Dobb's Journal, November 1985, Canada.
Interfacing Modula-2 to Assembler.

[4]. H.Mehrtens, David W.Smith
Anwendungsorientierte Modellierung für Produktionsplanung und -steuerung.
Springer-Verlag, 1985.

Software Tools Vol.3
von-Ernst-Knoll-Institut Hannover, Universität Hannover.

[5]. David Cohen
Pascal , Ada and Modula-2.
Byte, August 1984, pp. 215-232.

[6]. David Wagner
LOGIC Pascal Vs. Modula-2 a discerning look.
Byte, August 1984, pp. 420-430.

[7]. Jacques Mosser, Bernard
Prozeßsteuerung in Modula-2.
Springer-Verlag 1985.

[8]. Richard Eisenbeis
Modula-2 für Programmierer.
Carl Hanser Verlag 1985.

KOMUNIKATY I OFERTY

SYSTEM KOMPUTEROWY MX-16

MX-16 jest wielodostępnym systemem komputerowym średniej mocy obliczeniowej. Zbudowany na procesorze zgodnym z minikomputerem MERA-400 jest produkowany w następującej konfiguracji:

- procesor 16-bitowy;
- koprocesor zmiennoprzecinkowy (48-bitowy) i numeryczny (32-bitowy);
- pamięć operacyjna podstawowa o pojemności 256KB;
- pamięć dodatkowa o pojemności 8MB;
- pamięć stała inicjująco-diagnostyczna o pojemności 64KB;
- zegar czasu rzeczywistego;
- procesory peryferyjne MULTIX (jeden lub dwa), przez które można dołączyć maksymalnie 64 urządzenia :
  - terminale dowolnego typu (max. 64)
  - czytniki i perforatory taśmy papierowej (po 2 szt.);
  - drukarki z interface CENTRONICS lub LOGABAX (do 4 szt.);
  - pamięć taśmową PT-305/310 (do 8 przewijaków);
  - dyski twarde typu Winchester (20,40 lub 80MB) (do 8 szt.);
  - dyski elastyczne (360KB lub 1,2MB) (do 8 szt.).

Konstrukcja komputera MX-16 ma postać biurka z szufladą o wymiarach: 750 x 500 x 500 (mm).
Zasilacz o mocy 380W posiada układy zabezpieczające przed krótkotrwałymi zanikami sieci oraz układy podtrzymania zasilania elementów pamięciowych systemu. System (zarówno sprzęt jak i oprogramowanie) wyposażony jest w mechanizmy automatycznego restartu po zaniku zasilania. Jest to zasadnicza cecha umożliwiająca jego zastosowanie w środowiskach profesjonalnych, produkcyjnych, różniąca MX-16 od innych komputerów (w szczególności typu PC).

MX-16 jest systemem prawdziwie wielodostępnym, wielo-użytkownikowym i wieloprogramowym. Oznacza to, że w danym momencie może z komputera korzystać, nie przeszkadzając sobie wzajemnie, wielu równoprawnych użytkowników, z których każdy może uruchmiać wiele zadań (programów), tak jakby był jedynym użytkownikiem.

Aby można było uzyskać takie cechy systemu, zarówno sprzęt jak i system operacyjny, muszą mieć zaimplementowane niezbędne mechanizmy.

Celem umożliwienia wielodostępnej pracy sprzęt posiada następujące cechy:

- istnieją dwa tryby pracy procesora: uprzywilejowany i użytkowy. Tylko pod kontrolą systemu operacyjnego może się odbywać dostęp do zasobów systemu dzielonych przez wielu użytkowników;
- istnieje mechanizm ochrony pamięci uniemożliwiający jednym użytkownikom niszczenie pamięci innych użytkowników;
- istnieją mechanizmy służące do synchronizacji procesów, np. instrukcje semaforowe w procesorze;
- procesor jest dostatecznie wydajny, aby mieć czas na wykonywanie wielu programów;
- istnieją mechanizmy szybkiego przełączania stanu procesora pomiędzy różnymi zadaniami, a w szczególności pomiędzy programem użytkownika, a systemem operacyjnym;
- istnieją w systemie inne inteligentne moduły, które mogą wykonywać część pracy równocześnie z procesorem, odciążając go w ten sposób od czynności rutynowych;
- pamięć operacyjna posiada dostateczną pojemność i organizację (wirtualną) pozwalającą na rezydowanie w niej wielu wzajemnie chronionych przestrzeni, przydzielanych do różnych zadań. Nie trzeba więc wykonywać czasochłonnych operacji przeładowywania pamięci przy przełączaniu zadań, a procesor główny i procesory peryferyjne mogą pracować współbieżnie;
- wszystkie urządzenia służące do zapewnienia dostępu użytkownika do systemu mają taki sam interface sprzętowy i programowy, nie ma więc różnicy przy wykonywaniu zadania na dowolnym z nich. W szczególności nie istnieje jeden wyróżniony monitor (np. korzystający bezpośrednio z pamięci wewnętrznej procesora), na którym programy wykonują się zupełnie inaczej niż na pozostałych;
- sposób dołączania terminali do systemu jest elastyczny. Istnieje możliwość dołączenia terminali lokalnie przez proste łącza napięciowe (RS232C) lub odporne na zakłócenia łącza z optoizolacją (na odległości do 600m), lub też poprzez modemy. Szybkość transmisji w każdym łączu może być ustawiana indywidualnie.

Oprogramowanie systemowe posiada funkcje niezbędne dla realizacji wielodostępu;

- zasoby systemu, takie jak pamięć, urządzenia peryferyjne itp. mogą być przydzielane do różnych zadań w sposób gwarantujący ich "prywatność";
- inicjowanie równoczesnej pracy inteligentnych modułów systemu (procesorów) przy wykonywaniu jednego lub wielu zadań;
- szybkie przełączanie i aktywizacja zadań;
- elastyczna obsługa zróżnicowanych urządzeń, a w szczególności różnych terminali i urządzeń peryferyjnych;
- możliwość tworzenia rozproszonych systemów (sieci komputerowych).

Wszystkie te cechy posiada system komputerowy MX-16 pracujący pod systemem operacyjnym CROOK 5.

Żaden inny system komputerowy dostępny w kraju nie spełnia wszystkich wyżej wymienionych warunków. Są to cechy występujące tylko w tzw. superkomputerach. Można powiedzieć nawet, że zastosowany w systemie MX-16 oryginalny sposób organizacji pamięci operacyjnej i przyjęte mechanizmy wirtualizacji są jednymi z najbardziej efektywnych. Pozwala to na wykorzystywanie systemu MX-16 w zestawach wieloużytkownikowych z kilkudziesięcioma aktywnymi użytkownikami.

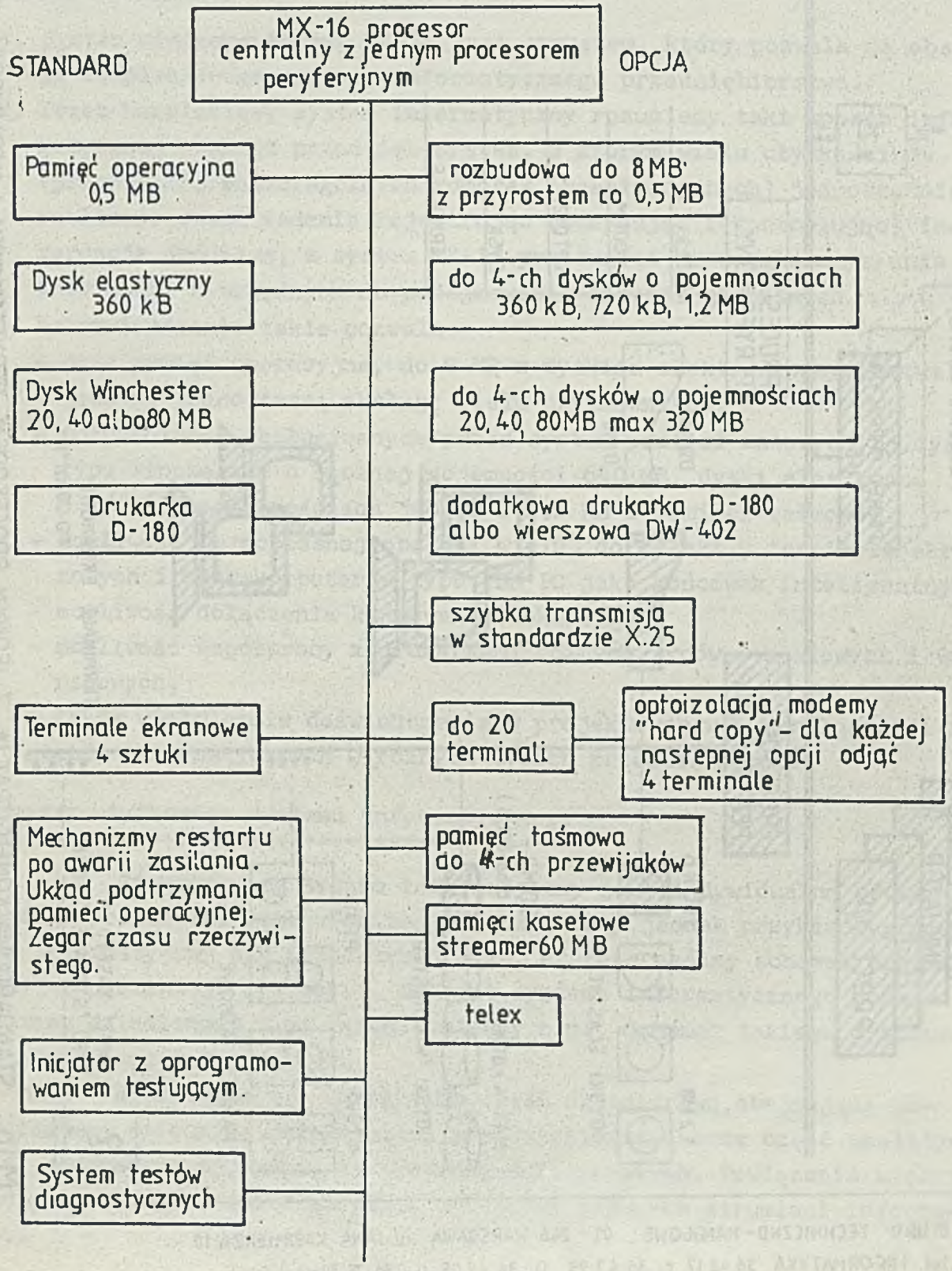
Inne systemy wielodostępne pozwalają na dołączenie tylko niewielkiej liczby użytkowników:

- najnowszy system IBM S2 w największym modelu (modelu 80 zrealizowanym na mikroprocesorze Intel 80386) może mieć maksymalnie do 8 użytkowników;
- system Fortune 16/32 zbudowany na mikrokroprocesorze Motorola 68000, może posiadać do 12 terminali, ale już przy 3 widać "zatykanie się" systemu i wyraźny spadek jego wydajności;
- komputery typu MicroVAX II również nie mogą pracować z więcej niż 16 użytkownikami.

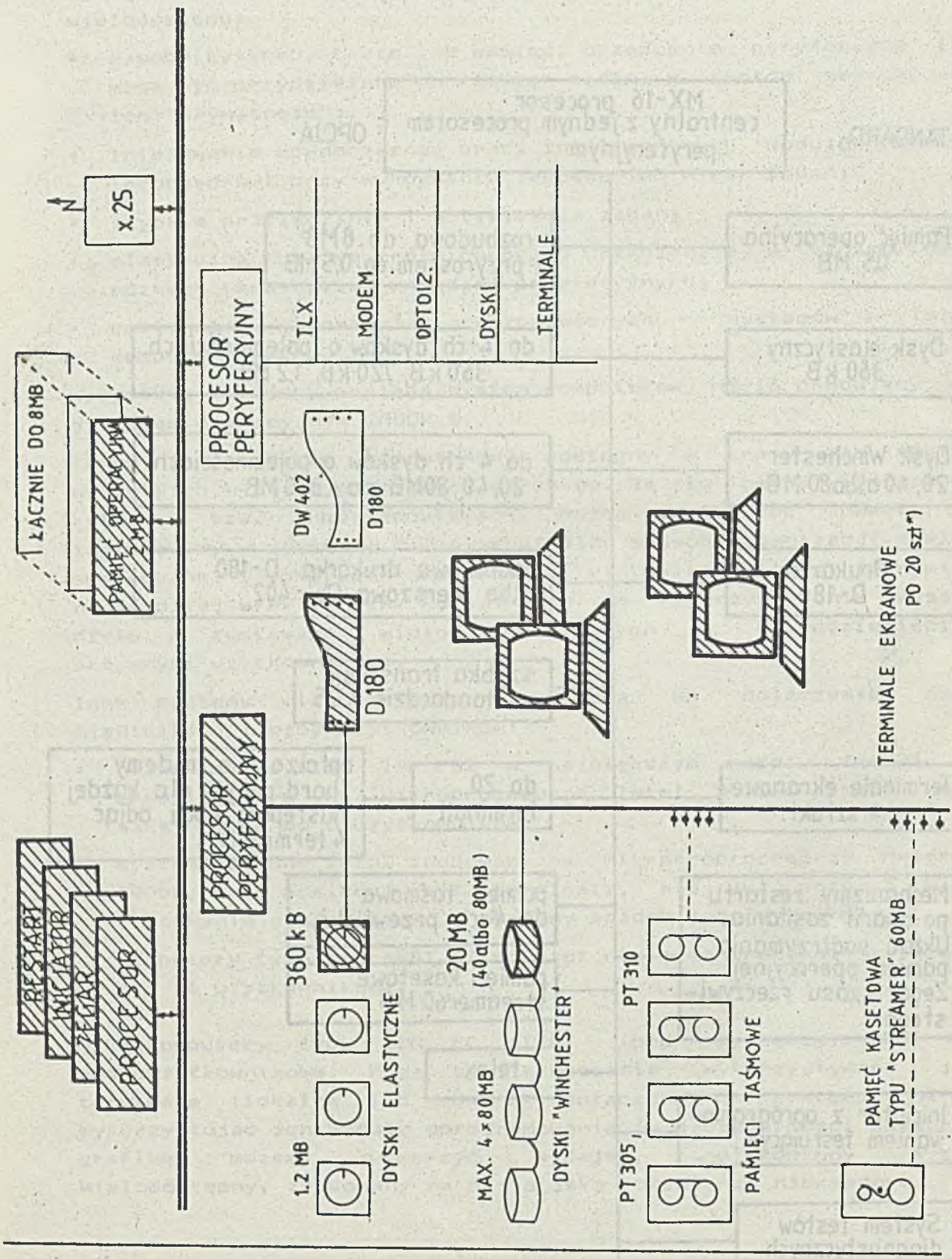
Mikrokomputery typu IBM PC XT/AT (pod systemem DOS) są jednoużytkownikowe. Mogą być skutecznie wykorzystywane jako terminale (lokalne lub zdalne) dołączone do systemu MX-16. Wykorzystując ich bogate oprogramowanie (w szczególności edytory i grafikę) możemy utworzyć wydajny, rozproszony system wielodostępny, zbudowany na MX-16 jako komputerze nadrzędnym.

amepol

PRZEDSIĘBIORSTWO ZAGRANICZNE



amepol
PRZEDSIĘBIORSTWO ZAGRANICZNE



M16: STANDARDOWA KONFIGURACJA Z JEDNYM PROCESOREM PERYFERYJNYM I MOŻLIWOŚCI JEJ ROZBUDOWY.

BIURO TECHNICZNO-HANDLOWE: 01-246 WARSZAWA ul. JANA KAZIMIERZA 10
tel.: INFORMATYKA 36 48 17, c. 36 47 93, DH 36 49 05, tlx. 81 25 39 apol.

Przedsiebiorstwo Zagraniczne
"A M E P O L"

SYSTEM MINIKOMPUTEROWY MX-16

O P R O G R A M O W A N I E U Ż Y T K O W E

Informacje ogólne

1. System minikomputerowy MX-16 jest sprzętem, który pozwala na obsługę kompleksowego systemu informatycznego przedsiębiorstwa.
2. Przez kompleksowy system informatyczny rozumiemy taki sposób informatycznej obsługi przedsiębiorstwa, w którym wielu użytkowników (pracowników poszczególnych komórek organizacyjnych) jednocześnie realizuje swoje zadania rejestrując (wczytując i kontrolując) informacje źródłowe, a system MX-16 przetwarza je w celu uzyskania informacji niezbędnych do podejmowania decyzji globalnych.
3. Na stwierdzenie takie pozwala:
   - duża pamięć operacyjna, do 8 MB z dyskiem elektronicznym,zezwalająca na jednoczesną obsługę wielu użytkowników,
   - różnorodność obsługiwanych przez system pamięci masowych,tj.dyski typu Winchester o łącznej pojemności 640 MB, dyski elastyczne 5 1/4 "o pojemnościach 360 kB i 1.2 MB i pamięci taśmowe,
   - możliwość jednoczesnej obsługi wielu, do 64 szt., terminali ekranowych i mikrokomputerów typu IBM PC jako końcówek inteligentnych,
   - możliwość dołączenia końcówek teleksowych,
   - możliwość współpracy z drukarkami różnych typów - znakowych i wierszowych,

   a także wieloletnie doświadczenia w projektowaniu i eksploatacji systemów informatycznych w różnych typach przedsiębiorstw.

Schemat użytkowego systemu informatycznego

Zintegrowany system informatyczny jest indywidualną cechą każdego przedsiębiorstwa.Poniżej przedstawiamy jednak przykładowy system informatyczny dla przedsiębiorstwa, którego ogólny schemat organizacyjny prezentuje rysunek 1. Schemat systemu informatycznego podaje systemy dziedzinowe i powiązania między nimi. Schemat takiego systemu przedstawiamy na rysunku 2.

W systemie tym wyróżniono część dziedzinową,obejmującą poszczególne dziedziny działalności przedsiębiorstwa oraz część analityczną stanowiącą nadbudowę do systemów dziedzinowych. Powiązania między systemami dziedzinowymi pokazują przepływy głównych strumieni informacji.

Na schemacie systemu wyróżniono powielarne systemy dziedzinowe które mogą być wdrażane do eksploatacji w dowolnym przedsiębiorstwie. Wskazano także te systemy indywidualne,które zostały opracowane dla różnych typów przedsiębiorstw (do przeniesienia możliwa jest metodyka opracowywania danego systemu).

W systemie operacyjnym CROOK-5, pod kontrolą którego realizowana jest eksploatacja systemów użytkowych, struktura systemu użytkowego jest wielopoziomowa (schemat graficzny przedstawia rysunek 3).

Poziom 0 jest poziomem systemu operacyjnego.Zapisana tu informacja jest dostępna dla użytkowników niższych szczebli - w sposób selektywny, stosownie do posiadanych przez nich uprawnień systemowych.

Poziom 1 jest najwyższym poziomem systemu użytkowego.Zapisana jest tu informacja o charakterze globalnym (zbiory główne, programy dostępne dla wybranych użytkowników z poziomów niższych). Dostęp do zapisanej tu informacji dla użytkowników niższych poziomów może być regulowany.

Poziom 2 jest poziomem głównych użytkowników systemu.Zapisane tu informacje są globalnymi w zakresie danej dziedziny, którą system obsługuje.

Poziom 3 jest poziomem pośrednim, grupującym użytkowników sprawujących funkcje nadzorcze w systemach dziedzinowych.Przechowywana jest tu informacja o charakterze lokalnie zbiorczym, np. transakcje z danego okresu.

Poziom 4 jest poziomem operatorów systemu, którzy prowadzą podstawową obsługę systemu (rejestracja informacji, sprawdzanie jej poprawności, proste aktualizacje zbiorów, itp).

Struktura systemu użytkowego w ramach systemu operacyjnego jest indywidualną cechą każdego przedsiębiorstwa.

Opis powielarnych systemów dziedzinowych

Jako powielarne systemy dziedzinowe traktujemy takie systemy, których zasady logiczne są identyczne we wszystkich przedsiębiorstwach - dostosowanie systemu polega na stworzeniu w danym przedsiębiorstwie odpowiedniej bazy indeksowej.

W naszym rozumieniu systemami powielarnymi są:

- system finansowo-kosztowy, prowadzący ewidencję transakcji gospodarczych przedsiębiorstwa i rozliczanie kosztów jego działalności,
- system "Środki Trwałe", prowadzący ewidencję i rozliczanie środków trwałych,
- system gospodarki materiałowej, obejmujący całokształt zagadnień związanych z obrotem materiałów,

- system kadrowy, obsługujący ewidencję kadrową przedsiębiorstwa.

A. System finansowo - kosztowy

System dopuszcza dwunastoznakową budowę poziomą symbolu konta analitycznego oraz praktycznie nieograniczoną liczbę kont w układzie dziesiętnym. Stwarza to możliwość rozbudowy analitycznej ewidencji księgowej wg potrzeb każdego użytkownika. System charakteryzuje:

- zasilanie systemu na wejściu z wszystkich zdarzeń gospodarczych podlegających ewidencji księgowej, bezpośrednio z tradycyjnych dokumentów źródłowych,
- prostota wdrożenia,
- dowolność cyklu przetwarzania,
- wysoki stopień automatyzacji technologii przetwarzania typu konwersacyjnego,
- obostrzone kryteria automatycznej kontroli programowej, tak każdego indywidualnego dowodu księgowego jak i łącznej sumy zbioru w powiązaniu z sumami kontrolnymi,
- funkcjonalność uzyskiwanych wyników podstawowych i wielowariantowych w postaci pełnej ewidencji analitycznej, pełnej ewidencji syntetycznej, pomocniczej ewidencji analitycznej dla potrzeb analizy kont rozrachunkowych, analitycznego dokumentu z wyniku automatycznego rozliczenia kosztów, dodatkowych dowolnych informacji w zmiennych przekrojach wg potrzeb, kalkulacji jednostkowej podziałowej-prostej.

Wdrożenie systemu jest krótkotrwałe, nie dopuszcza się dublowania w okresie wdrożenia z ewidencją tradycyjną. Prawidłowość wdrożenia jest warunkowana przez:

- opracowanie zakładowego planu kont z uwzględnieniem wymagań rachunkowości i systemu,
- zapoznanie pracowników księgowości z nowymi warunkami ich pracy, wynikającymi z kryteriów i wymogów techniczno-technologicznych systemu w zakresie obiegu dokumentów, techniki ich kontroli wstępnej, merytorycznej, formalnej i rachunkowej, numerowania dokumentacji, techniki dekretacji i ich rejestracji.

B. Sytem gospodarki środkami trwałymi.

System obejmuje zagadnienia ewidencji środków trwałych w przedsiębiorstwie w ujęciu wielozakładowym, wraz z automatycznym comiesięcznym naliczaniem umorzeń i amortyzacji oraz ich rozliczaniem za okresy miesięczne i okresy narastające w roku wg grup rodzajowych i stanowisk kosztów oraz ewentualnych innych podziałów wg potrzeb.

Prowadzona przez system kartoteka środków trwałych zawiera pełny zestaw informacji dotyczących poszczególnych środków trwałych (obiektów) - do 30 informacji o każdym - w tym takie jak: grupa rodzajowa, numer inwentarzowy, miejsce powstawania kosztów, nazwa obiektu i jego opis, data przyjęcia do eksploatacji, wartość inwentarzowa brutto i netto, umorzenie, amortyzacja i inne.

Zmiany w kartotece wprowadzane są w oparciu o typowe dokumenty obrotu i zmian w ruchu środków trwałych (OT, LT, PT, MT) z terminala ekranowego w trybie konwersacyjnym i są na bieżąco ewidencjonowane przez system. Wszystkie zrealizowane transakcje zapisywane są do zbioru transakcji i w dowolnym czasie mogą być dostępne do wglądu lub wydruku.

Możliwe jest też korygowanie niektórych informacji w zbiorze głównym w trybie tzw. transakcji doraźnych. W trybie tym możliwe jest wyświetlenie na monitorze lub wydruk dowolnych informacji zawartych w kartotece.

System zapewnia komplet danych i informacji niezbędnych do sporządzenia obowiązującej sprawozdawczości GUS.

C. System gospodarki materiałowej.

System obejmuje zagadnienia ewidencji stanów i obrotów materiałów i przedmiotów nietrwałych w użytkowaniu oraz rozliczenia kosztów zużycia materiałów. W systemie prowadzona jest ilościowa kartoteka magazynowa, ilościowo-wartościowa kartoteka stanów i obrotów magazynowych oraz rozliczenia kosztów zużycia materiałów wg miejsc powstawania kosztów.

Zmiany w kartotece magazynowej wprowadzane są bezpośrednio z dokumentów obrotu magazynowego (Pz,Rw,Zw,Wz,Mm,..) i na bieżąco są ewidencjonowane przez system.

Użytkownikom systemu (służby zaopatrzenia,księgowość, służba magazynowa) dostarczana jest pełna informacja dotycząca gospodarowania materiałami, a mianowicie:

- pełna, ilościowo-wartościowa, ewidencja stanów i obrotów magazynowych,
- informacja o stanie zapasów w magazynach i odchyleniach od ustalonych w przedsiębiorstwie normatywów,
- pełne rozliczenie kosztów zużycia materiałów i przedmiotów nietrwałych w użytkowaniu,
- rozliczenia spisów magazynowych z natury,
- dane do obowiązującej sprawozdawczości GUS.

Konwersacyjna forma komunikacji użytkownika z systemem umo-

żliwia jego obsługę przez służby zaopatrzenia i księgowości, bez potrzeby znajomości zagadnień informatycznych.

D. System kadrowy.

System obejmuje prowadzenie ewidencji kadrowej pracowników przedsiębiorstwa. W systemie prowadzona jest pełna informacja o wszystkich pracownikach zatrudnionych w przedsiębiorstwie.

System zapewnia dostęp do informacji o pracowniku upoważnionym służbom funkcjonalnym przedsiębiorstwa oraz niezbędne informacje do sporządzenia obowiązującej sprawozdawczości GUS.

Oprogramowanie wspomagające tworzenie systemów użytkowych.

W systemie minikomputerowym MX-16 dostępne jest oprogramowanie narzędziowe wspomagające tworzenie własnych systemów użytkowych. Są to
- kompilator języka R A L, specjalizowanego na przetwarzanie danych,
- oprogramowanie BD´87, dotyczące tworzenia systemów użytkowych w języku FORTRAN.
- ponadto Pascal, "C", Basic, Asembler

A. Kompilator języka R A L

Specjalizowany interpreter zleceń przetwarzania danych R A L (Random Access Language) służy do tworzenia oprogramowania użytkowego z zakresu przetwarzania danych i umożliwia:
- łatwy i przejrzysty podział zbiorów na rekordy i pola oraz zachowanie sporządzonego opisu w zbiorach specjalnych,
- szybki dostęp do dowolnej informacji zapisanej w zbiorach dyskowych,
- jednoczesny dostęp do dowolnej liczby zbiorów,
- synchronizację dostępu do zbiorów z dowolnej liczby końcówek,
- automatyczną rejestrację danych z częściową kontrolą oraz możliwością ustalania i poprawiania rejestrowanych pól,
- dokonywanie operacji arytmetycznych i znakowych na polach różnych typów oraz konwersję typów,
- definiowanie dowolnej liczby zmiennych, tablic, podprogramów i funkcji zewnętrznych,
- wygodną i elastyczną organizację pętli,
- wygodne formatowanie wydruków.

Łącznie z interpreterem zleceń przetwarzania danych dostarczana jest biblioteka funkcji wewnętrznych interpretera oraz biblioteka programów wspomagających tworzenie i uruchamianie programów własnych.

B. Oprogramowanie BD'87

System organizacji i operowania na zbiorach danych BAZA DANYCH BD'87 jest oprogramowaniem służącym do tworzenia własnych systemów użytkowych w języku FORTRAN. Zawiera programy do tworzenia i obsługi zbiorów danych oraz procedury fortranowskie pozwalające na korzystanie z informacji zawartych w zbiorach danych w programach własnych użytkownika.

W ramach systemu BD'87 działają następujące programy:

- BAZA - program tworzenia i modyfikacji struktur zbiorów danych,
- DANE - standartowy program do wpisywania,wyświetlania,poprawiania i usuwania danych ze zbiorów,
- DRUK - program do wykonywania wydruków. Program tworzy opis wydruku oraz posiada możliwości drukowania danych w sposób selektywny oraz drukowania sum częściowych i zbiorczych,a także sum średnich,
- SORT - program porządkowania zawartości zbiorów danych wg dowolnie wybranego klucza sortowania (rosnąco lub malejąco). Klucz sortowania może się składać z wielu pól (danych) wchodzących w skład dokumentu.
- LACZ - program do łączenia zbiorów o różnej strukturze danych - działa na strukturach zbiorów i na danych.

Procedury fortranowskie są wykorzystywane w programach własnych użytkownika i umożliwiają przetwarzanie danych zawartych w zbiorach. Są to następujące procedury:

OTWORZ i ZAMKNIJ - rozpoczynające i kończące współpracę ze zbiorem danych,

CZYTNAST i PISZNAST - pozwalające na zapis i odczyt informacji w trybie sekwencyjnym,

CZYTAJ i PISZ - pozwalające na odczyt i zapis informacji w trybie bezpośrednim,

DOPISZ i USUN - procedury dopisywania i usuwania rekordu ze zbioru

PORZ - procedura sortowania informacji w zbiorach danych

Oprogramowanie BD'87 pozwala na łatwe tworzenie własnych systemów użytkowych i umożliwia szybką ich konserwację.

Wykaz użytkowników systemów minikomputerowych MX-16

Systemy minikomputerowe MX-16 wykorzystywane są do obsługi informatycznej w wielu przedsiębiorstwach, wyższych uczelniach, biurach projektów i innych instytucjach na terenie całego kraju. Tworzone tam systemy informatyczne znajdują się w różnych fazach kompletno-

ści. Najbardziej znaczącymi ośrodkami są:

1. Stocznia Remontowa "RADUNIA" w Gdańsku
2. Huta Szkła Okiennego "SZCZAKOWA" w Jaworznie
3. Zakłady Elektroniczne "UNIMOR" w Gdańsku
4. Gdańskie Zakłady Nawozów Fosforowych w Gdańsku
5. Wytwórnia Sprzętu Komunikacyjnego "PZL-GORZYCE" w Gorzycach k/Sandomierza
6. Zakłady Mięsne w Gdyni
7. Huta Szkła Walcowanego "JAROSZOWIEC" w Jaroszowcu k/Olkusza
8. Kluczewskie Zakłady Papiernicze w Kluczach
9. Zespół Opieki Zdrowotnej w Zakopanem
10. Częstochowskie Zakłady Przemysłu Lniarskiego "WARTA" w Częstochowie
11. Zakłady Osprzętu Samochodowego POLMO w Łodzi
12. Zakłady Automatyki Chemicznej "METALCHEM" w Gliwicach
13. Śląskie Przedsiębiorstwo Konstrukcji Stalowych "MOSTOSTAL" w Zabrzu
14. Zakłady Zmechanizowanego Sprzętu Domowego "PREDOM-ZELMER" w Rzeszowie
15. Żyrardowskie Zakłady Tkanin Technicznych w Żyrardowie
16. Ośrodek Badawczo-Rozwojowy Maszyn Włókienniczych "POLMATEX-CENARO" w Łodzi
17. Urząd Miejski w Sopocie
18. Odlewnia Żeliwa ZM "URSUS" w Gorzowie Wlkp.
19. Olsztyńskie Zakłady Opon Samochodowych "STOMIL" w Olsztynie
20. Huta Szkła Okiennego "SANDOMIERZ" w Sandomierzu

Usługi programowania systemów użytkowych

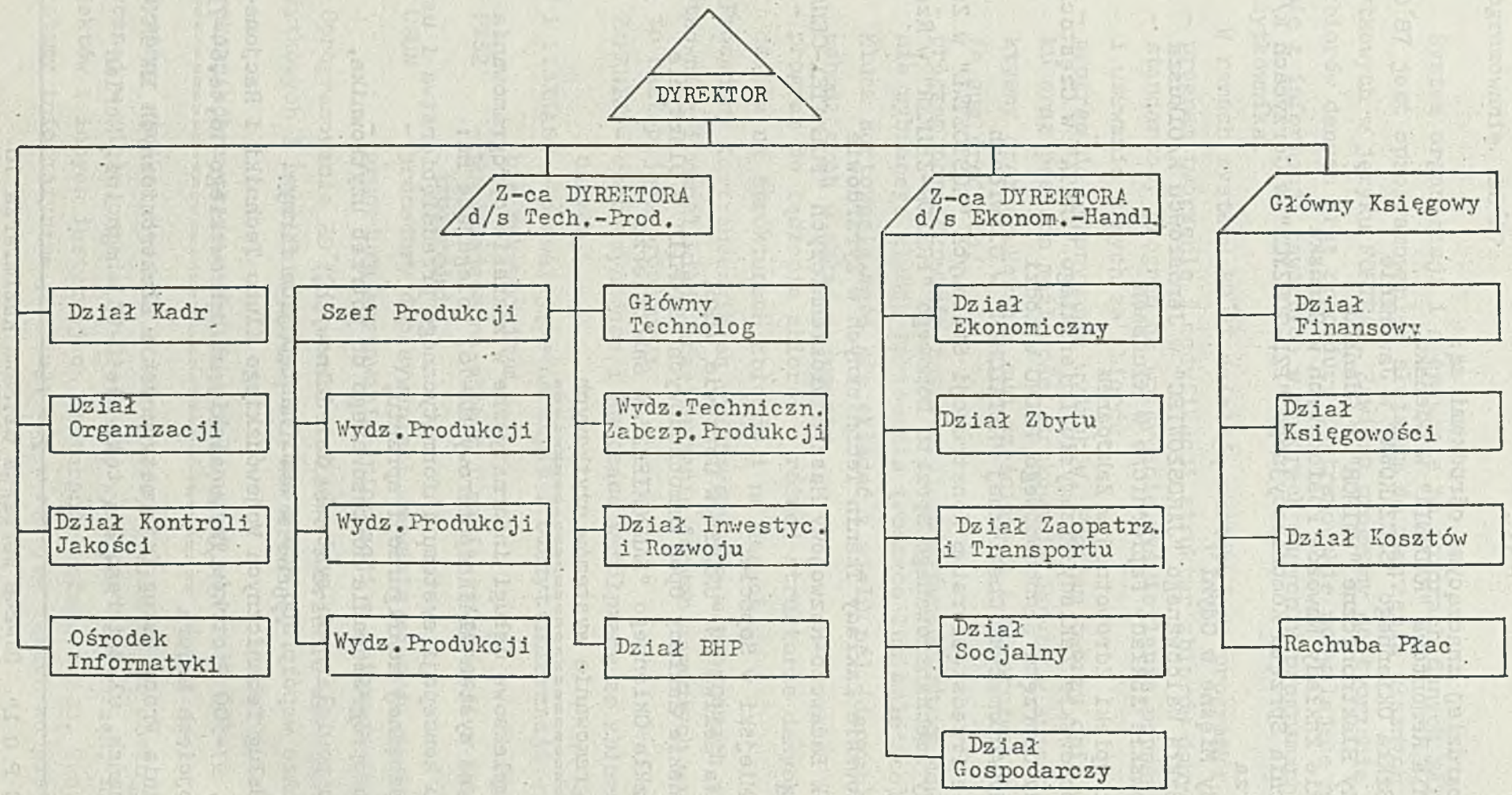
Kompleksowe usługi informatyczne w zakresie oprogramowania użytkowego na system minikomputerowy MX-16, polegające na:

- tworzeniu koncepcji systemu informatycznego przedsiębiorstwa i ustalaniu niezbędnej konfiguracji sprzętu,
- tworzeniu oprogramowania dostosowanego do potrzeb użytkownika,
- wdrażaniu powielarnych systemów dziedzinowych,

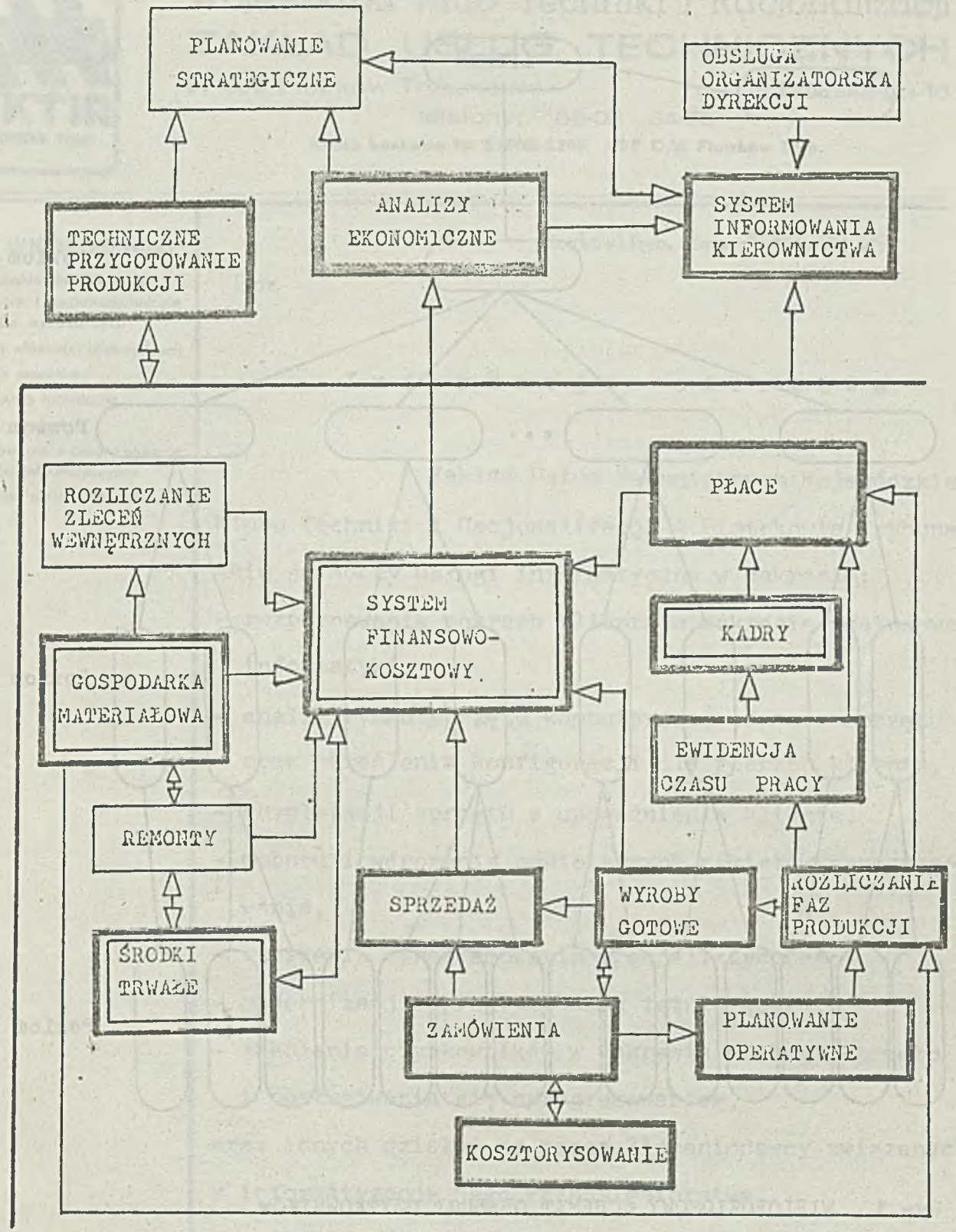
świadczą także współpracujące z nami następujące firmy:

1. Zakład Usług Technicznych Wojewódzkiego Klubu Techniki i Racjonalizacji, 97-300 Piotrków Trybunalski, ul.Dąbrowskiego 16,te.64-75 - mgr Wojciech Badura
2. Spółdzielnia Pracy Usług Informatycznych i Elektronicznych INFORMATYK 80-251 Gdańsk, ul.Batorego 21, tel. 41-10-17 - mgr inż. Marian Porzych

---



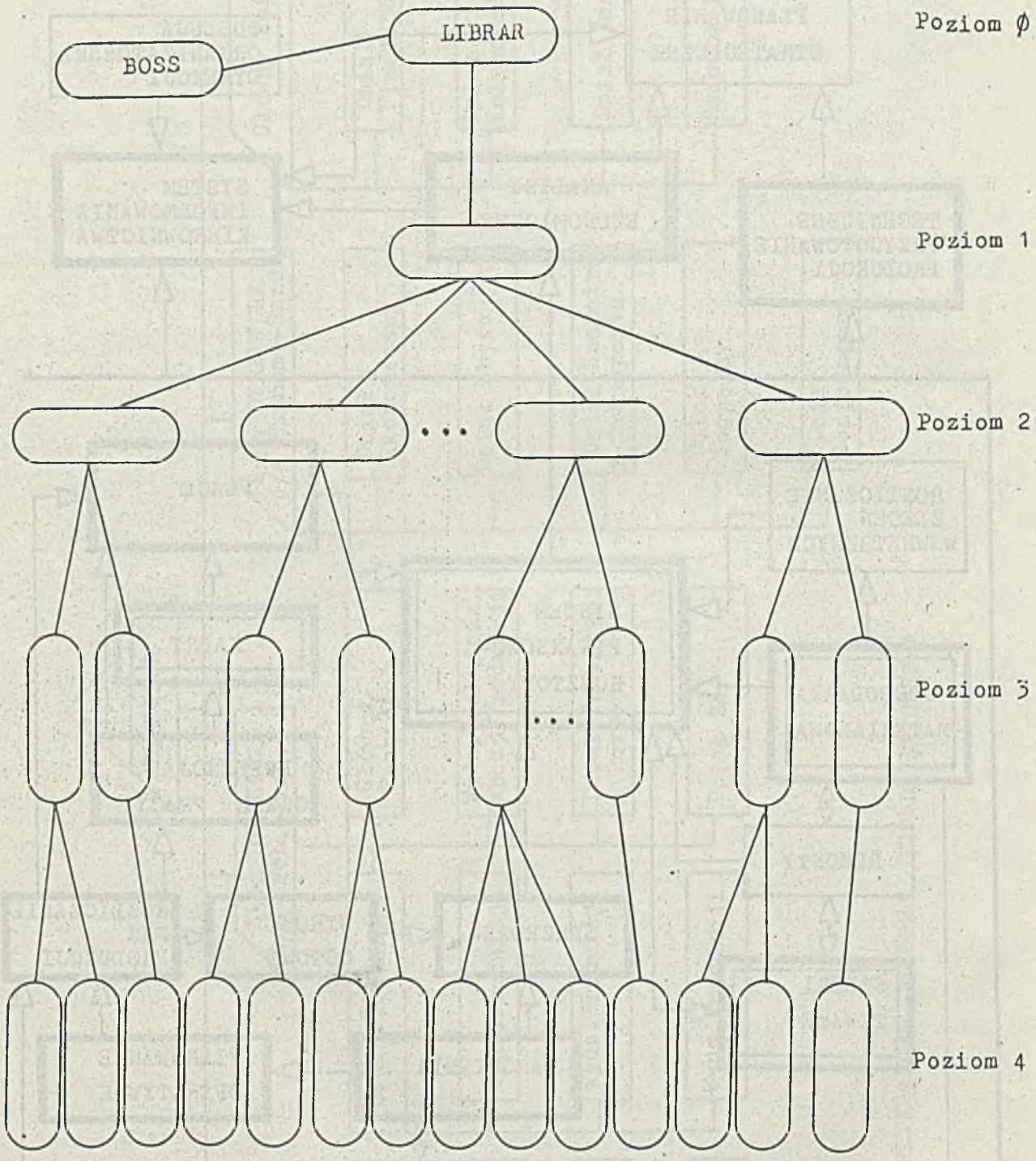
Rys.1. PRZYKŁADOWY SCHEMAT PRZEDSIĘBIORSTWA



Rys.2. SCHEMAT PRZYKŁADOWEGO SYSTEMU INFORMATYCZNEGO

SYSTEM POWIELARNY DO WDROŻENIA W DOWOLNYM PRZEDSIĘBIORSTWIE

SYSTEMY REALIZUJĄCE PODOBNE FUNKCJE SĄ EKSPLOATOWANE NA MX-16



Rys.3. WIELOPOZIOMOWY SCHEMAT OBSŁUGI UŻYTKOWNIKÓW

Wojewódzki Klub Techniki i Racjonalizacji
ZAKŁAD USŁUG TECHNICZNYCH
97-300 Piotrków Trybunalski ul. Dąbrowskiego 16
telefony: 65-03 64-75
Konto bankowe Nr 59008-5395 NBP O/W Piotrków Tryb.

ZUT WKTIR oferuje:

- opracowania dokumentacji, wdrożenia i rozpowszechnianie projektów wynalazczych
- ochrony własności przemysłowej
- badania patentowe
- tłumaczenia techniczne
- opinie i ekspertyzy
- projektowanie i programowanie systemów informatycznych
- obliczanie efektów ekonomicznych

Piotrków Tryb., dnia 1987.09.29

L.dz.

Informacja ofertowa

Zakład Usług Technicznych Wojewódzkiego Klubu Techniki i Racjonalizacji w Piotrkowie Trybunalskim świadczy usługi informatyczne w zakresie:

- rozpoznawania potrzeb klienta w zakresie zastosowań informatyki,
- analiz rynku sprzętu komputerowego, doboru sprzętu oraz określenia konfiguracji dla potrzeb klienta,
- kompletacji sprzętu z upoważnienia klienta,
- doboru i wdrożenia powielarnych pakietów oprogramowania,
- tworzenia oprogramowania specjalistycznego,
- modernizacji oprogramowania istniejącego,
- szkolenia użytkowników w zakresie obsługi sprzętu i posługiwania się oprogramowaniem,

oraz innych działań na rzecz Zleceniodawcy związanych z informatyzacją jego Przedsiębiorstwa.

Wszelkich informacji udziela mgr Wojciech Badura.

Zapraszamy do współpracy!!!

752 — Sp. Pr. „Introl." P-ków — 1000 A4

LISTA UCZESTNIKÓW
KONFERENCJI UŻYTKOWNIKÓW MINIKOMPUTERA MERA-400

1. ADAMCZAK Lucyna, Huta Szkła Okiennego "SANDOMIERZ", ul. Portowa 24, 27-600 Sandomierz, tel.sł. 3041, wew. 143.
2. ADAMCZYK Zbigniew, Z-d Informatyki Spółdzielni Mieszkaniowych, Spółdzielnia Osób Prawnych, ul. Kliny 2, 31-465 Kraków, tel.sł. 11-94-67.
3. BADURA Wojciech, Zakład Usług Technicznych WKTiR, ul. Dąbrowskiego 16, 97-300 Piotrków Trybunalski, tel.sł. 64-75/276-42.
4. BARAN Marek, Wytwórnia Filtrów "PZL Sędziszów", ul. Fabryczna 4, 39-120 Sędziszów Młp., tel. sł. 32 wew. 270/184.
5. BARCZYK Stanisław, Huta Szkła Walcowanego "Jaroszowiec", ul. Kolejowa 1, 32-312 Jaroszowiec, tel.sł. 309-14, wew. 143.
6. BAUER Zbigniew, Zakład Informatyki Spółdzielni Mieszkaniowych, Spółdzielnia Osób Prawnych, ul. Kliny 2, 31-465 Kraków, tel.sł. 11-94-67.
7. BESTYNSKI Piotr,Poznańskie Biuro Projektów Budownictwa Przemysłowego, Ratajczaka 10/12, 60-567 Poznań, tel.sł. 622-31.
8. BIEŃKOWSKI Andrzej, Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne, ul. Zwycięstwa 140, 75-613 Koszalin, tel.sł. 277-51 wew. 121.
9. BOBCOW Andrzej, Stocznia Remontowa"Radunia", ul. Na Ostrowiu 1, 80-873 Gdańsk, tel. sł. 31-68-31.
10. BONIECKI Marek, Woj. Związek Spółdzielni Pracy, Ośrodek Obliczeniowy, ul. Dyrekcyjna 5, 80-852 Gdańsk. tel.sł. 20-82-15.
11. BRANIECKI Andrzej, Instytut Okrętowy, Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-16-43.
12. BRZÓSKA Elżbieta, INTiBS PAN, Pl. Katedralny 1, 50-950 Wrocław, tel.sł. 22-10-71 wew. 72.
13. CICHOCKI Mieczysław, BPBBO , Miastoprojekt-Łódź, ul. Traugutta 21/23, 90-950 Łódź, tel.sł. 33-81-19.

14. CIELECKI Leszek, Woj. Ośrodek Inf., ul. Partyzantów 74, 80-254 Gdańsk,
15. CIOTUCHA Teodor, Fabryka Maszyn Górniczych "Pioma", ul. Żarskiego 38, 97-300 Piotrków Trybunalski, tel.sł. 404-90 wew. 297, dom. 40510.
16. CŁAPA Józef, Fabryka Maszyn Górniczych "PIOMA", ul. Żarskiego 38, 97-300 Piotrków Trybunalski, tel.sł. 404-90 wew. 297, dom. 27-808.
17. CZERNIAK Zbigniew, Politechnika Gdańska, Instytut Okrętowy, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-17-08,
18. DOMINIKOWSKI Antoni, Politechnika Warszawska, Instytut Elektrotechniki Teoretycznej i Miernictwa Elektrycznego, ul. Koszykowa 75, 00-661 Warszawa, tel.sł. 21-00-7328.
19. DRZYMAŁA Aldona, Centrum Komputeryzacji Rynku, Stary Rynek 97/100, 61-773 Poznań, tel.sł. 222-802.
20. DŻOGA Jerzy, Przedsiębiorstwo Zagraniczne "Amepol", ul. Jana Kazimierza 10, 01-248 Warszawa, tel.sł.36-48-17, 36-49-05.
21. FORSZPANIAK Arleta, Ośrodek Obliczeniowy WZSP, ul. 27 Grudnia 3, 61-737 Poznań, tel.sł. 573-61 wew. 18,28.
22. GACKI Jerzy, Centrum Komputeryzacji Rynku "CEKAR", Stary Rynek 97/100, 61-773 Poznań, tel.sł. 222-802.
23. GERYSZEWSKI Andrzej, Przedsiębiorstwo Zagraniczne "AMEPOL", ul. Jana Kazimierza 10, 01-248 Warszawa, tel.sł.36-48-17.
24. GOŁĄB Andrzej, Wytwórnia Sprzętu Komunikacyjnego "PZL-Gorzyce", 39-432 Gorzyce k/Sandomierza, tel.sł. 3451- wew. 450.
25. GRALL Jerzy, Instytut Włókiennictwa, ul. Brzezińska 1/3, 90-950 Łódź, tel. sł. 33-96-00 wew. 214.
26. GRALL Tadeusz, Centralny Ośrodek Badawczo-Rozw. Masz. Włók. ul. Wólczańska 55/59. 80-950 Łódź, tel.sł. 32-85-70 wew. 243.
27. GRUSZKA Jerzy, Huta Szkła Okiennego "Sandomierz", ul. Portowa 24, 27-600 Sandomierz, tel.sł. 3041 wew. 229.
28. HAUKE Roman, Żyrardowskie Zakłady Tkanin Technicznych, ul. Okrzei 51, 96-300 ŻYRARDÓW, tel.sł. 20-31, wew. 284.
29. IZDEBSKI Mirosław, Instytut Okrętowy, Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-14-31.
30. JABŁONSKI Witold, Gdańskie Zakłady Nawozów Fosforowych, ul. Kujawska 2, Gdańsk, tel.sł. 438-310, dom. 31-21-21.

31. KALINSKA Danuta, Instytut Melioracji i Użytków Zielonych w Falentach, 05-550 RASZYN, tel.sł. 500531 wew. 243.
32. KAPAŁA Zenon, Instytut Okrętowy, Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł. 47-18-08.
33. KORDYS Marian, Częstochowskie Zakłady Przemysłu Lniarskiego "WARTA", ul. Krakowska 45, 42-201 Częstochowa, tel.sł. 332. 426-31.
34. KOREŃ Dominik, Woj. Ośrodek Infor. ul. Partyzantów 74, 80-254 Gdańsk.
35. KRAWCZYK Zdzisław, CBS i PBW "Hydroprojekt" O/Gdańsk, ul. Powstańców Warszawskich 202, Gdańsk-Wrzeszcz, tel.sł. 41-20,61, 41-20-92, 41-60-80.
36. KRUSZYNSKA Mirosława, Instytut Meteorologii i Gosp.Wodnej, ul. Podleśna 61, 01-673 Warszawa, tel.sł. 34-54-66.
37. KRYCZKA Jan, Politechnika Łódzka, Instytut Inżynierii Chemicznej, ul. Wólczańska 175, 90-924 Łódź, tel.sł. 365522 wew. 715.
38. KULIK KULIKOWSKA Elżbieta, Akademia Wychowania Fizycznego, Zakład Metod i Analiz Matematycznych, ul. Marymoncka 34, Warszawa, tel.sł. 34-04-31, wew. 322.
39. KURZYK Aleksander, Woj.Ośrodek Informatyki, ul. Partyzantów 74, 80-254 Gdańsk.
40. KWAPINSKI Henryk, Zakłady Automatyki Chemicznej "Metalchem", ul. Chorzowska 44, 44-101, Gliwice, tel.sł. 31-64-41.
41. LABUDA Roman, Wytwórnia Sprzętu Komunikacyjnego "PZL-Gorzyce", 39-432 Gorzyce k/Sandomierza, tel.sł. 3451 wew. 450.
42. LASOTA Jarosław, Gdańskie Zakłady Nawozów Fosforowych, ul. Kujawska 2, Gdańsk, tel.sł. 438-310, dom. 53-13-05.
43. ŁUKASZEWSKA Małgorzata, Ośrodek Obliczeniowy WZSP, ul. 27 Grudnia 3, 61-737 Poznań, tel.sł. 573-61, wew.28,18.
44. MACIUK Bronisław, Politechnika Śląska, Wydział Metalurgiczny, Katedra Organizacji Produkcji, ul. Krasińskiego 8, Katowice, tel.sł. 516671 wew. 24.
45. MADZGAŁA Jerzy, Hutnicze Przedsiębiorstwo Remontowe w Gliwicach, Zakład Nr 2, ul. Zwycięstwa 14, 44101 Gliwice, tel.sł. 31-00-21-24 wew. 235.
46. MALOWANA Wiesława, Rybnicki Zakład Prefabrykacji PPPW "PREFBET", ul. Wiejska 7, Rybnik, tel.sł. 264-51.

47. MARTIN Włodzimierz, Politechnika Gdańska, Instytut Okrętowy, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł. 47-26-31.
48. MELLER Adam, Woj. Komenda Straży Pożarnych, ul. Gen. Bema 17, 82-300 Elbląg, tel.sł. 276-91, wew. 213.
49. MICHALEC Janusz, Zespół Opieki Zdrowotnej, ul. Kamieniec 10, 34-501 Zakopane, Pracownia Informatyki, tel.sł. 50-64 wew. 288.
50. MICHAŁOWSKI Henryk, Inst. Mel. i Użytków Zielonych w Falentach, 06-550 Raszyn, tel.sł. 500-531 wew. 243.
51. NIEMIEC Józef, Ośrodek Inform. CZSP , ul. Bonifraterska 14, Warszawa, tel.sł. 31-41-63, 31-68-72.
52. NIKODEMSKI Marek, Instytut Okrętowy, Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł. 47-18-08.
53. NOWACKI Kazimierz,CBSiPBW HYDROPROJEKT, ul. Dubois 9, 00-182 Warszawa, tel.sł. 387041 wew. 246, dom. 33-18-76.
54. NOWAK Piotr, Instytut Chemii Organicznej i Fizycznej, Politechnika Wrocławska, ul. Wybrzeże St. Wyspiańskiego 27, 50-370 Wrocław, tel.sł. 20-24-85/219-274.
55. OLESKI Janusz, BPBBO-MIASTOPROJEKT-Łódź, ul. Traugutta 21/23, 90-950 Łódź, tel.sł. 33-81-19.
56. OLKOWSKI Henryk, Instytut Ciężkiej Syntezy Organicznej "Blachownia", ul. Energetyków 9, 47-225 Kędzierzyn-Koźle, tel.sł. 332-41 wew. 5459.
57. OŁUBIEC Waldemar, Woj. Ośrodek Informatyki, ul. Partyzantów 74, 80-254 Gdańsk.
58. PAJĄCZKOWSKI Janusz, Wojewódzkie Przedsiębiorstwo Energetyki Cieplnej, ul. Dzierżyńskiego 5, 85-315 Bydgoszcz, tel.sł. 342-81, wew. 239.
59. PEREK Maria, Huta Szkła Okiennego "Szczakowa", ul. Kolejarzy 81 32-520 JAWORZNO, tel.sł. 774-41, wew. 120.
60. PIECHOTA Ryszard, Gdańskie Zakłady Elektroniczne "UNIMOR", ul. Rzeźnicka 54/56, 80-822 Gdańsk, tel.sł. 375-330.
61. PIETRZAK Krystyna, Zakład Ekonomiki i Informatyki , MPK, ul. Piotrkowska 147/149, 90-440 Łódź, tel.sł. 364812/528070.

62. PISIEWICZ Andrzej, Wojewódzkie Biuro Projektów Zabrze, ul. Wolności 286, 41-800 Zabrze, tel.sł. 71-20-21 wew.84.
63. PIWKO Elżbieta, Główne Biuro Studiów i Projektów Przeróbki Węgla "Separator", ul. Armii Czerwonej 2, 40-952 Katowice, tel.sł. 58-60-71 wew. 513.
64. PŁACZEK Dorota, Rybnicki Zakład Prefabrykacji PPPW "PREFBET", ul. Wiejska 7, Rybnik, tel.sł. 264-51.
65. PYZIAK Krystyna, Przedsiębiorstwo Zagraniczne "AMPEL", ul. Jana Kazimierza 10, 01-248 Warszawa, tel. sł. 36-48-17.
66. RATYNSKA Elżbieta, Instytut Melioracji i Użytków Zielonych, 05-550 Falenty - Raszyn, tel.sł. 500531 wew. 274.
67. REWILAK Zygmunt, Zakład Informatyki Spółdzielni Mieszkaniowych, Spółdzielnia Osób Prawnych, ul. Kliny 2, 31-465 Kraków, tel.sł. 11-94-67.
68. ROSŁON Waldemar, Spółdzielnia "Otwock", ul. Świerczewskiego 5A, 05-402 Swider,
69. SMOL Anna, Zespół Opieki Zdrowotnej, Pracownia Informatyki, Kamieniec 10, 34-500 Zakopane, tel.sł. 508-64 wew. 288.
70. SOKOŁOW Wojciech, CBSiPBW HYDROPROJEKT, ul. Dubois 9, 00-182 Warszawa, tel.sł. 387041 wew. 259,
71. STARZYNSKI Zdzisław, Instytut Meteorologii i Gospodarki Wodnej, ul. Podleśna 61, 01-673 Warszawa, tel.sł. 34-54-66.
72. STECURA Jadwiga, Przedsiębiorstwo Spedycji Krajowej, Zarząd, ul. Ordona 2a, 01-287 Warszawa, tel.sł. 365107.
73. STEFANIAK Tomasz, Biuro Projektów Przemysłu Lekkiego "BEDETE", Pl. Zwycięstwa Nr 2, 90-950 Łódź, tel.sł. 74-48-78 wew. 114 lub 113.
74. SZAL Bogdan, Gdańskie Zakłady Elektroniczne "UNIMOR",, ul. Rzeźnicka 54/56, 80-822 Gdańsk, tel.sł. 375-330.
75. SZARATA Danuta, Wojewódzkie Przedsiębiorstwo Energetyki Cieplnej, ul. Dzierżyńskiego 5, 85-315 Bydgoszcz, tel.sł. 342-81 wew. 239.
76. SZCZEPANIAK Barbara, Biuro Projektów Budownictwa Komunalnego, ul. Tuwima 22/26,90-002 Łódź, tel.sł. 32-32-75/48-08-45.
77. SZCZĘSNY Ryszard, Żyrardowskie Zakłady Tkanin Technicznych, ul. Okrzei 51, 96-300 Żyrardów, tel.sł. 20-31, wew. 284.

78. ŚCIĄŻKO Marek, Kluczewskie Zakłady Papiernicze, woj. Katowice, 32-310 Klucze, Katowice tel. 518-889 wew. 1272 lub Olkusz 31360 wew. 1272.
79. TARNAWSKA Anna, Przedsiębiorstwo Zagraniczne "AMEPOL", ul. Jana Kazimierza 10, 01-248 Warszawa, tel.sł.36-48-17.
80. TRUSEWICZ Franciszek, Olsztyńskie Zakłady Opon Samochodowych "STOMIL", Al. Zwycięstwa 71, 10-950 Olsztyn, tel.sł. 33-12-79.
81. WERBINSKI Ryszard, Gdańskie Zakłady Nawozów Fosforowych, ul. Kujawska 2, 80-958 Gdańsk, tel.sł. 438-268 dom. 380-181.
82. WĘGRZYN Jan, Politechnika Warszawska, Instytut Elektrotechniki Teoretycznej i Miernictwa Elektr. Wydz. Elektr. ul. Koszykowa 75, Warszawa, tel.sł. 21-007-525.
83. WIERZBICKI Jan, Huta Szkła Okiennego "Szczakowa", ul. Kolejarzy 81, 32-520 Jaworzno, tel.sł. 774-41 wew. 120.
84. WIŚNIEWSKA Maria, Biuro Projektów Budownictwa Morskiego BIMOR, pl. Batorego 4, 70-207 Szczecin, tel. sł. 403-374.
85. WITKO Jan, Państwowy Ośrodek Maszynowy, ul. Wodociągowa 1, 66-500 Strzelce Krajeńskie, tel.sł. 914.
86. WOŹNIAK Urszula, Stocznia Remontowa "Radunia", Dział Planowania Informatyki, ul. Na Ostrowiu 1, 80-873 Gdańsk, tel.sł. 31-68-31.
87. ZAMYSŁOWSKI Edward, Fabryka Osprzętu Samochodowego POLMO, ul. Przybyszewskiego 99, 93-126 Łódź, tel.sł. 81-41-40, 81-42-06 wew. 388.
88. ZAWADZKI Wiesław, Zakłady Mechaniczne "Gorzów", ul. Przemysłowa 14/15, 66-400 Gorzów Wlkp. tel.sł. 27221 wew. 314.
89. ZIELINSKI Stefan, Politechnika Gdańska, Instytut Okrętowy, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-14-24.
90. ZIELONKA Jerzy, BPBBO MIASTOPROJEKT, ul. Więckowskiego 20, 90-722 Łódź, tel.sł. 32-81-00 wew. 36.
91. ZIÓŁKOWSKI Andrzej, Instytut Badań Systemowych PAN, ul. Newelska 6, 01-447 Warszawa, tel.sł. 361901 wew. 220, dom. 339796.
92. ŻYŁŁA Romuald, Politechnika Łódzka, Instytut Inżynierii Chemicznej, ul. Wólczańska 175, 90-924 Łódź, tel.sł. 365522 wew. 837 lub 843.

Lista Członków Porozumienia Użytkowników Minikomputera MERA-400

Lp.	Nazwa i adres instytucji	Pełnomocnik	Tel.
1	2	3	4
1	Akademia Górniczno-Hutnicza, Al. Mickiewicza 30, 30-059 Kraków		
	- Instytut Automatyki	mgr inż. Janusz Miller	33-81-00 w.2855
	- Instytut Elektroniki	dr inż. Marek Książkiewicz	33-81-22 w. 30-22
	- Instytut Metalurgii Metali Nieżelaznych	dr inż. Janusz Dominiak	tel.wew.26-86
	- Instytut Maszyn Hutniczych i Automatyki	mgr inż. Wojciech Słota	tel.wew.30-32
2	Akademia Wychowania Fizycznego Z-d Metod i Analiz Matemat. ul. Marymoncka 34, 01-813 Warszawa	Wiesław Ostaszewski	34-04-31 wew. 322
3	Biuro Projektów Budownictwa Komunalnego ul. Grunwaldzka 2, 82-300 Elbląg	mgr Piotr Czajkowski	259-03
4	Biuro Projektów Służby Zdrowia ul. Astrów 10, Katowice 40-045	mgr Wiesław Nowak	582-841
5	Biuro Projektów Budownictwa Ogólnego "MIASTOPROJEKT", ul. Kraszewskiego 36, Kraków 30-110	Krzysztof Orlof	225100 w.385
6	Biuro Projektowo-Badawcze Budownictwa Ogólnego MIASTOPROJEKT, ul. Traugutta 21/23 90-113 Łódź	Mieczysław Cichocki	33-81-19
7	Biuro Projektowo-Badawcze Budownictwa Ogólnego MIASTOPROJEKT-2, ul. Więckowskiego 20, Łódź 90-722	dr inż. L. Maro	32-81-00 w.36
8	Biuro Projektów Budownictwa Komunalnego ul. Tuwima 22/26, 90-002 Łódź	mgr inż. Barbara Szczepaniak	32-32-75

1	2	3	4
9	Biuro Projektów Przemysłu Mięsnego ul. Chocimska 28, 00-957 Warszawa	mgr inż. Barbara Jung	49-80-10
10	Centralny Ośrodek Badawczo-Rozwojowy Maszyn Włókienniczych POLMATEX-CENARO ul. Wólczańska 55/59, 90-950 Łódź	mgr inż. Tadeusz Grall	32-85-70 w.243
11	Centralne Biuro Studiów i Projektów Budownictwa Wodnego HYDROPROJEKT ul. Powstańców Warszawskich 202 80-162 Gdańsk	mgr inż. Henryk Noga	41-20-61
12	Częstochowskie Zakłady Przemysłu Lniarskiego WARTA, ul. Krakowska 45, Częstochowa 42-200	Marian Kordys	426-31
13	Fabryka Maszyn Górniczych "PIOMA", ul. Żarskiego 38, Piotrków Trybunalski 97-300	Teodor Ciotucha	404-90
14	Gdańskie Z-dy Nawozów Fosforowych, ul. Kujawska 2,80-958 Gdańsk,	Ryszard Werbiński	438-263
15	Gdańskie Z-dy Elektr. "UNIMOR" ul. Rzeźnicka 54/56 , 80-822 Gdańsk	Jerzy Biesiadowski	375-330
16	Huta Szkła Okiennego "SZCZAKOWA" ul. Kolejarzy 81, 32-520 Jaworzno	mgr inż. Jan Wierzbicki mgr inż. Maria Perek	774-41 w. 120 774-41 w. 282
17	Huta Szkła Walcowanego "JAROSZOWIEC" ul. Kolejowa 1, 32-312 Jaroszpwiec	mgr inż. St. Barczyk	309-14 w. 143
18	Huta Szkła Okiennego "SANDOMIERZ" ul. Portowa 24, 27-600 Sandomierz	Jan Kocój Józef Kucharski	30-41 w. 143
19	Instytut Fizyki Molekularnej PAN ul. Smoluchowskiego 17/19,60-179 Poznań	dr Wojciech Malinowski dr Andrzej Dezor	67-40-71 w. 264 w. 219
20	Instytut Włókiennictwa, ul. Gdańska 91/93 90-950 Łódź	mgr Jerzy Grall mgr Krystyna Białaa	339-600 w. 214 w. 175
21	Instytut Melioracji i Użytków Zielonych Falenty, 05-550 Raszyn	mgr Henryk Michałowski mgr inż. Sławomir Mączyński	500-531 w. 243

1	2	3	4
22	Komenda Wojewódzka Straży Pożarnych ul. Bema 17/21, 82-300 Elbląg	Adam Meller	276-91 w. 213
23	Krakowskie Biuro Projektowo_Badawcze Budownictwa Ogólnego, Al. Powstańców Warszawy 10, 31-549 Kraków	mgr inż. Anna Dymek	11-87-33 w. 347
24	Kluczewskie Zakłady Papiernicze 32-310 Klucze	mgr inż. Marek Sciążko	518-889 w. 1272
25	Miejskie Przedsiębiorstwo Komunikacyjne ul. Piotrkowska 147/149, 90-440 Łódź	mgr Krystyna Pietrzak	52-80-70
26	Okręgowy Szpital Kolejowy ul. Medyków 5, 40-760 Katowice	mgr inż. Marian Pustelnik	525-011 w. 22
27	Ośrodek Badawczo-Rozwojowy Systemów Automatyki ul. Czerwonej Armii 66/72, 61-807 Poznań	dr inż. Andrzej Wachowski	53-437
28	Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne ul. Zwierzyniecka 10, 60-813 Poznań	B. Biełkowska	472-41 w. 27
29	Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne ul. Pobożnego 5, 70-508 Szczecin	inż. Michał Umiastowski	434-88
30	Olsztyńskie Z-dy Opon Samochodowych "STOMIL" Al. Zwycięstwa 71, 10-575 Olsztyn	Kazimierz Majewski	33-07-41 w. 568
31	PKP Oddz. Geodezyjny, ul. Roździeńskiego 1, Katowice 40-202	inż. Jerzy Gotwald	57-55-97
32	Polskie Górnictwo Naftowe i Gazownictwo Górnośląski Z-d Gazownictwa ul. Gwardii Ludowej 11, 41-800 Zabrze	mgr inż. Tadeusz Kochel	71-52-21 w. 54-15
33	Przedsiębiorstwo Zagraniczne "AMEPOL" Pl. Żelaznej Bramy 1, 00-136 Warszawa	Jerzy Dżoga	20-34-75
34	Politechnika Łódzka 90-924 Łódź		
	- Instytut Maszyn Przepływowych ul. Stefanowskiego	Roman Malinowski	36-55-22 w.12-52
	- Inst. Inż. Chemicznej ul. Wólczańska 175,	mgr inż. Jan Kryczka	36-55-22 w.715

1	2	3	4
35	Politechnika Warszawska Instytut Elektroenergetyki ul. Koszykowa 75, 00-662 Warszawa	mgr inż. Andrzej Cichy	21-86-88
	Politechnika Warszawska Wydział Budownictwa i Maszyn Rolniczych ul. Łukasiewicza 17, 09-400 Płock	Marek Malinowski	260-61 w. 227
36	Politechnika Gdańska Instytut Okrętowy, ul. Majakowskiego 11/12 80-952 Gdańsk	mgr inż. Andrzej Braniecki	47-16-43
37	Politechnika Wrocławska ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław	Piotr Nowak	20-24-12
38	Śląska Akademia Medyczna ul. Medyków, 40-752 Katowice	dr inż. Janusz Pietkiewicz	527-081 w. 14-66
39	Śląskie Przedsiębiorstwo Konstrukcji Stalowych i Urządzeń Przemysłowych "MOSTOSTAL" ul. Wolności 191, 41-800 Zabrze	Zbigniew Urbaniec	71-32-21 w. 201
40	Stocznia "RADUNIA" ul. Na Ostrowiu 1, 80-873 Gdańsk	Urszula Woźniak	31-68-31
41	Spółdzielnia Mieszkaniowa Zakład Informatyki ul. Kliny 2, 31-485 Kraków	Zbigniew Bauer	11-94-67
42	Warszawska Spółdzielnia Mieszkaniowa ul. Krasińskiego 16, 01-581 Warszawa	Zdzisław Jagodziński	39-51-00
43	Wojewódzkie Biuro Projektów ul. Wolności 286, 41-801 Zabrze	mgr inż. Karol Bresler	71-20-21 w. 10
44	Wojewódzkie Przedsiębiorstwo Energetyki Cieplnej, ul. Dzierżyńskiego 5, 85-315 Bydgoszcz	Ewa Konecka	342-81 w. 239
45	Wytwórnia Sprzętu Komunikacyjnego PZL-Gorzyce, 39-432 Gorzyce	Andrzej Gołąb	34-51 w. 450

1	2	3	4
46	Zakład Automatyki Chemicznej "METALCHEM" ul. Chorzowska 44, 44-101 Gliwice	inż. H. Kwapiński	31-64-41, w. 102
47	Zespół Opieki Zdrowotnej ul. Kamieniec 10, 34-500 Zakopane	mgr inż. Anna Smol	20-21 w. 288
48	Zakłady Sprzętu Zmechaniznowanego "ZELMER" Rzeszów		
49	Zakład Usług Technicznych WKTiR ul. Dąbrowskiego 16 97-300 Piotrków Trybunalski	mgr Wojcieoh Baduna	11-94-67
50	Żyrardowskie Zakłady Tkanin Technicznych ul. Okrzei 51, 96-300 Żyrardów	Ryszard Szczęsny	20-31
51	Wojewódzki Ośrodek Informatyki ul. Partyzantów 74, 80-254 Gdańsk	Dominik Koreń	
52	Wojewódzki Związek Spółdzielczości Pracy 80-852 Gdańsk , Dyrekcyjna 5	Krzysztof Górski	20-82-15
53	Bydgoskie Zakłady Przemysłu Gumowego "STOMIL", ul. Toruńska 155, 85-950 Bydgoszcz	mgr inż. Henryk Stachowski	
54	Centralne Biuro Studiów i Projektów Budownictwa Wodnego "Hydroprojekt", ul. Dubois 9, 00-182 Warszawa,	mgr inż. Kazimierz Nowacki	38-70-41 w. 246

Prorem 170/89/120