

P. 1877/74

6

1974

informatyka

SPIS TREŚCI NR 5/74

	Dyskowy system operacyjny dla komputerów JS EMC <i>Ryszard Tryba</i>	1
	Wykorzystanie tablic decyzyjnych w programowaniu oraz rozwiązywaniu problemów typu projektowego <i>Benedykt Burgielski</i>	7
	Perspektywy powszechnej komputeryzacji w USA <i>Andrzej Targowski</i>	12
	W sprawie elementarnego wykładu informatyki <i>Jan M. Szymański</i>	17
	O wpływie komputerów na społeczeństwo <i>Joseph Weizenbaum</i>	29
	Programista jako nawigator — oprac. <i>St. Matwin</i>	35
TRYBUNA CZYTELNIKA	Metodyka stosowania programów etykietujących — <i>Zygmunt Topolewski</i>	21
Z KRAJOWEGO BIURA INFORMATYKI I ZJEDNOCZENIA INFORMATYKI	I Sesja Programistów Zjednoczenia Informatyki — <i>R. Wiśniewski</i>	22
Z KRAJU	INFORMATYKA z wizytą w MERAMACIE — <i>D. Prawdź</i> Czas teraźniejszy i problemy przyszłości ZETO Wrocław — <i>K. Bernatowicz</i>	38 39
	Informacje różne	43
ZE ŚWIATA	Przedstawiamy firmę BENSON — <i>M. Hołyński</i>	44
	Informacje różne	45
ÓŚRODKI INFORMATYKI PREZENTUJĄ	ETOCHEM Mały System Informatyczny Centrali Zjednoczenia — <i>Z. Bieńko</i>	46
PRZEGLĄD WYDAWNICTW	Z księgarni książek importowanych — <i>J. Adamus</i> Bibliografia wydawnictw polskich z dziedziny informatyki — <i>J. Klamborowski</i>	48 37 III okł.
	Ogłoszenia	IV okł.

KOLEGIUM REDAKCYJNE

Redaktor Naczelny prof. dr hab. Leon ŁUKASZEWICZ

prof. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), mgr inż. Marek HOŁYŃSKI, Władysław KLEPACZ, Elżbieta KOŁODZIEJSKA, doc. dr hab. Antoni MAZURKIEWICZ, inż. Dorota PRAWDZIC (zast. redaktora naczelnego), doc. dr inż. Andrzej TARGOWSKI

Sekretarz Redakcji mgr Krystyna WROŃSKA

Red. tech. Józef DUSZA

RADA PROGRAMOWA

Przewodniczący — Prof. dr hab. Andrzej STRASZAK



WYDAWNICTWA
CZASOPISM
TECHNICZNYCH
NOT

Warszawa
Czackiego 3/5

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pokój 331, tel. 27-71-40 lub centrala 28-82-61 w. 285, dyżury redakcji 10,00-13,00

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka”. Z. 2. Zam. 240. Papier druk. sat. IV kl. 70 g 61 × 86. Obj. 4 ark. druk. Nakład 5960 + 30. W-47

Cena egzemplarza zł 8.—

INDEKS 36707

Prenumerata roczna zł 96.—

Informatyka

dawniej Maszyny Matematyczne
zastosowania w gospodarce, technice i nauce



Nr 6

MIESIĘCZNIK

1974

ROK X

Czerwiec

16 LIP 1974

ORGAN KRAJOWEGO BIURA INFORMATYKI I POLSKIEGO KOMITETU AUTOMATYCZNEGO
PRZETWARZANIA INFORMACJI NACZELNEJ ORGANIZACJI TECHNICZNEJ



Rejestr czasop.
Nr 744/k

Mgr RYSZARD TRYBA, urodzony we Wrocławiu w r. 1946, ukończył studia matematyczne na Uniwersytecie Wrocławskim (1969) i rozpoczął pracę w Ośrodku Badawczo-Rozwojowym przy ELWRO. Początkowo zajmował się oprogramowaniem maszyn cyfrowych serii ODRA 1300, a obecnie kieruje pracownią oprogramowania podstawowego Jednolitego Systemu EMC.

P. 1877 / 74

RYSZARD TRYBA

Ośrodek Badawczo-Rozwojowy
WZĘ ELWRO
Wrocław

681.327.63.06



Dyskowy system operacyjny dla komputerów JS EMC

Omówiono części składowe DOS/JS, mianowicie: programy sterujące oraz programy przetwarzające łącznie z językami programowania (ASSEMBLER, FORTRAN, PL/1 i RPG) i programami usługowymi. Przedstawiono również i sposób generowania DOS/JS.

W skład dyskowego systemu operacyjnego DOS/JS wchodzi translatory i programy usługowe pracujące pod kontrolą programu sterującego. DOS/JS wymaga jednostki centralnej spełniającej standardy jednolitego systemu, pamięci operacyjnej o minimalnej pojemności 32K bajtów i co najmniej jednej jednostki pamięci dyskowej.

DOS/JS został opracowany z myślą o osiągnięciu następujących celów:

- skrócenie czasu potrzebnego na zaprogramowanie i rozwiązanie problemu
- zwiększenie ogólnej wydajności ośrodków obliczeniowych
- ułatwienie pracy personelowi ośrodków obliczeniowych (przede wszystkim programistom i operatorom).

DOS/JS pozwala użytkownikom:

- tworzyć biblioteki programów w pamięci dyskowej, przy czym programy mogą być zapisane w bibliotece w postaci źródłowej, wynikowej (tzn. po ich przetłumaczeniu przez translator) lub w postaci gotowej do uruchomienia (tj. po zredagowaniu ich przez program łączący)
- dzielić programy źródłowe na moduły, z których każdy może być zaprogramowany w dowolnym języku, a następnie połączyć te moduły w jeden program

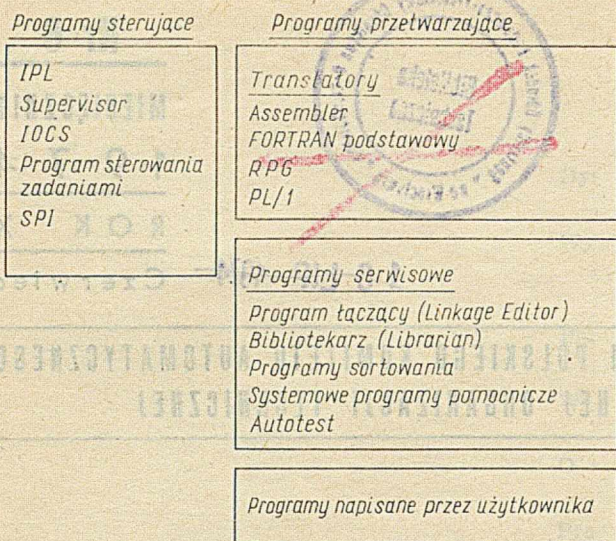
- zastosować nakładanie programu, tzn. dzielenie programu na tzw. fazy, z których tylko niektóre, konieczne w danym momencie przetwarzania, znajdują się w pamięci operacyjnej, natomiast reszta programu przechowywana jest w pamięci dyskowej. Pozwala to redukować wielkość pamięci operacyjnej potrzebnej do wykonania dużych programów

- testować i poprawiać programy
- tworzyć programy niezależne od adresów urządzeń i kanałów oraz — przy zachowaniu określonych warunków — niezależne od typu wykorzystanych urządzeń.

Rys. 1. przedstawia zawartość dyskowego systemu operacyjnego. Po lewej stronie wymienione są składniki części sterującej systemem, po prawej — programy przetwarzające. Do programów przetwarzających zaliczone są translatory, programy usługowe, jak również wszystkie programy napisane przez użytkownika. System sterowania wykorzystuje większość programów przetwarzających operacjami wejścia/wyjścia (IOCS), natomiast wszystkie programy korzystają z funkcji programu nadzorczego (Supervisor).

Programy sterujące

Ażeby uzyskać zadawalającą efektywność systemu przetwarzania danych, pewne prace muszą być automatycznie sterowane, co pozwala uniknąć, ewentualnie skrócić przestoje, występujące na przykład przy wprowadzaniu programów do pamięci operacyjnej lub przy zmianie taśm magnetycznych. Efektywne wykonywanie kolejnych zadań zabezpieczone jest przez:



Rys. 1. Składowe dyskowego systemu operacyjnego

- stale znajdujący się w pamięci operacyjnej program nadzorczy (Supervisor)
- umieszczenie w bibliotekach często wykorzystywanych programów (lub części programów)
- sterowanie wykonaniem ciągu zadań, którą to funkcję spełnia program sterujący zadaniami (Job Control).

Tabela. Podział pamięci operacyjnej

Specjalne informacje i rejony komunikacji	Strefa programów sterujących
Część główna programu nadzorczego, w skład której wchodzi między innymi następujące procedury: — obsługi przerwań — organizacji przesłań zewnętrznych — wprowadzania faz — analizy błędów — ochrony pamięci (niezbędna przy wieloprogramowaniu) — obsługi zegara oraz tablice wejścia/wyjścia	
Obszar wymienny przeznaczony na następujące procedury: — obsługi błędów — otwierania zbiorów — zamykania zbiorów — komunikacji operator-system	Strefa programów problemowych
Drugoplanowy obszar programowy (BG), w którym programy pracują w reżimie wsadowym (pod kontrolą programu sterowania zadaniami)	
Pierwszoplanowy — 2 obszar programowy (F2), w którym programy mogą pracować w reżimie wsadowym lub w reżimie pojedynczych programów.	
Pierwszoplanowy — 1 obszar programowy (F1), w którym programy mogą pracować w reżimie wsadowym lub w reżimie pojedynczych programów.	

Tabela I przedstawia schemat wykorzystania pamięci operacyjnej w systemie DOS/JS.

Wieloprogramowość

System DOS/JS umożliwia wieloprogramowość tylko ze stałymi obszarami programowymi, tzn. liczba obszarów i ich wielkości są z góry ustalone. Ponieważ obszarów programowych może być co najwyżej trzy, DOS dopuszcza podział czasu wykorzystania jednostki centralnej na co najwyżej trzy programy (BG, F2 i F1). Liczbę i wielkość obszarów programowych wyznacza się podczas generowania systemu. Wielkości obszarów mogą być zmieniane przez operatora (interwencja zewnętrzna), a nie przez sam program sterujący.

Każdy program zajmuje spójny obszar pamięci operacyjnej.

Programy drugo- i pierwszoplanowe. Przy wieloprogramowości rozróżnia się programy problemowe dwóch rodzajów: programy drugoplanowe (BG-programy) i programy pierwszoplanowe (F-programy).

BG-programy są uruchamiane przez program sterowania zadaniami zgodnie z opisem zadań (a więc jedynie w przetwarzaniu wsadowym), przy czym w jednym kroku zadania uruchamiany jest jeden program (jedno lub wielofazowy).

F-programy mogą być uruchamiane pojedynczo przez operatora, który podaje odpowiednie informacje inicjatorowi pojedynczych programów lub też w przypadku, gdy zestaw komputera jest odpowiedni (obszar pierwszoplanowy ma pojemność co najmniej 10 K i do dyspozycji jest dostateczna liczba urządzeń wejścia/wyjścia) — przez program sterowania zadaniami.

A więc przy odpowiednim zestawie istnieją możliwości przetwarzania wsadowego również i w pierwszoplanowych obszarach programowych. Rodzaj przetwarzania w pierwszoplanowych obszarach programowych określa się podczas generowania systemu. BG-programy i F-programy „zaczynają się” i „kończą” niezależnie jeden od drugiego i są logicznie jeden od drugiego niezależne. System może jednocześnie obsługiwać program drugoplanowy i jeden lub dwa programy pierwszoplanowe.

Priorytety nadawane są programom automatycznie z chwilą ich wprowadzenia do pamięci operacyjnej i są one determinowane przez obszary, w których te programy będą pracować. Najwyższy priorytet nadawany jest programowi znajdującemu się w obszarze F1. Niższy priorytet ma program z F2, a najniższy program — z BG.

Po obsłudze przerwania program nadzorczy przekazuje sterowanie temu z programów (będących w stanie gotowości do pracy), który ma najwyższy priorytet. Jeżeli żaden z programów nie jest w stanie gotowości (na przykład wszystkie programy czekają na zakończenie rozpoczętych transmisji), to system przechodzi w stan oczekiwania na przerwanie. Oprócz minimalnej pamięci 32K system wieloprogramowy wymaga istnienia ochrony pamięci (poprzez klucze pamięci). Określenie przetwarzania wsadowego

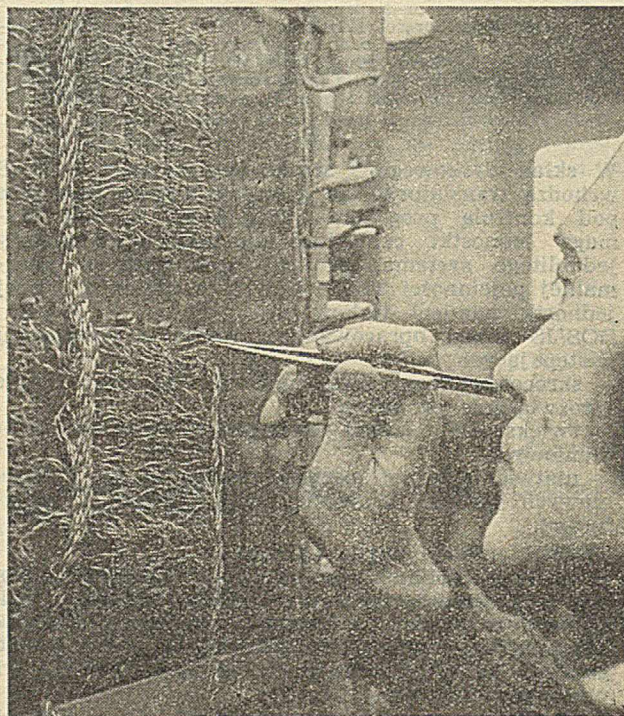


Foto 1. Zakłady ELWRO. Inż. Kurek kontroluje połączenia w jednostce centralnej R-30

dowego w pierwszoplanowych obszarach powoduje zarezerwowanie dla tych obszarów osobnych rejonów komunikacji. Programy korzystające z tych rejonów nie mogą być uruchamiane przy pomocy inicjatora pojedynczych programów.

Program nadzorczy (Supervisor)

Program nadzorczy przyjmuje i obsługuje przerwania wszystkich typów, inicjuje operacje wejścia/wyjścia oraz kontroluje ich przebieg. Zajmuje się wykrywaniem i analizą błędów i warunków wyjątkowych, wprowadzaniem do pamięci operacyjnej faz programów oraz wyborem programów do wykonania. Program nadzorczy zawiera również wszystkie tablice niezbędne do pracy systemu.

W celu oszczędności pamięci operacyjnej część procedur programu nadzorczego nie znajduje się na stałe w pamięci operacyjnej, ale w bibliotece i jest wprowadzana do obszaru wymiennego z chwilą konieczności ich wykonania. Są to tzw. procedury wymienne. Spełniają one funkcje niezbyt często wykonywane, jak na przykład wypis zawartości pamięci operacyjnej, analiza błędów operacji wejścia/wyjścia (z wyjątkiem operacji dotyczących dysków — procedura ta znajduje się zawsze w pamięci operacyjnej). Procedurami wymiennymi są również procedury otwierające i zamykające zbiory, czy organizujące komunikację operator — system.

DOS/JS przewiduje programy dwóch rodzajów: przesuwalne i nieprzesuwalne. Programy przesuwalne mogą być poprawnie wykonywane niezależnie od miejsca pamięci operacyjnej, w którym się znajdują. Programy nieprzesuwalne mogą pracować tylko w określonym i stałym obszarze pamięci operacyjnej. Większość programów problemowych, jak również część główna programu nadzorczego są nieprzesuwalne.

Procedury wymienne są przesuwalne. Złożoność programu nadzorczego jest zależna od konfiguracji zestawu komputera. Dla każdego konkretnego zestawu tworzy się — podczas generowania systemu — odpowiedni program nadzorczy.

Komunikacja operator — system

Ważną cechą systemu DOS/JS jest prosta obsługa operatorska. Informacje dla operatora są wyprowadzane na standardowe urządzenie łączności z operatorem (najczęściej jest to monitor).

System żąda odpowiedzi krótkich, najczęściej jedno-wyrazowych np.: IGNORE (ignoruj) czy CANCEL (skasuj). W wielu przypadkach operator może nie podawać sprezywanej odpowiedzi, powierzając systemowi podjęcie decyzji.

Operator może przy pomocy procedury komunikacji z systemem zażądać:

- przerwania aktualnie wykonywanego zadania (na przykład w przypadku pętli)
- zatrzymania pracy systemu
- zapisu znacznika taśmy (TAPE MARK) na taśmie magnetycznej lub jej przewinięcia
- zmiany przydziału i przypisania urządzeń zewnętrznych
- podania aktualnej konfiguracji zestawu komputera
- zmiany wielkości obszarów programowych
- uruchomienia inicjatora pojedynczych programów.

Za pomocą procedury komunikacji operator — system można zmienić standardy ustalone na przykład w momencie generowania systemu. Dotyczy to przede wszystkim przepisania urządzeń zewnętrznych i wielkości obszarów programowych.

Program sterowania zadaniami (Job Control)

Program sterowania zadaniami steruje wykonywaniem ciągu zadań przy przetwarzaniu wsadowym. Każde zadanie dzieli się na tak zwane kroki, przy czym jeden krok oznacza wykonanie jednego programu. Opis zadania — przygotowany w postaci zdań sterujących dla programu sterowania zadaniami —

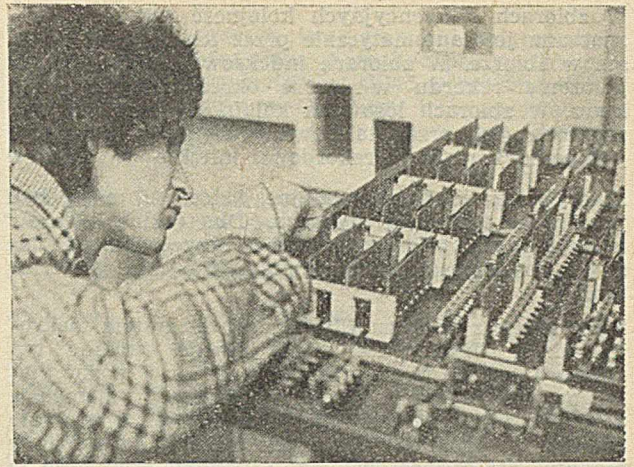


Foto 2. Inż. R. Chybicki dokonuje pomiarów elementów jednostki centralnej R-30. w Zakładach ELWRO

zawiera wszystkie niezbędne informacje potrzebne do poprawnego wykonania programów. Tak więc określa on, jakie konkretne urządzenia zewnętrzne będą używane przez programy, identyfikuje zbiory wykorzystane w zadaniu oraz określa, w jaki sposób ma odbywać się tłumaczenie i redagowanie programów.

Ze względu na to, że konkretne zestawy komputera różnią się między sobą konfiguracją (i adresacją) urządzeń w celu uniezależnienia (w możliwie maksymalny sposób) programów od konkretnego zestawu, wprowadzono tzw. logiczne jednostki urządzeń wejścia/wyjścia. W programach urządzenia są identyfikowane właśnie poprzez te jednostki, natomiast w opisie zadania przypisuje się tym jednostkom konkretne urządzenia fizyczne. Takie przypisanie jest ważne tylko na okres jednego zadania (a więc jest to przypisanie czasowe), po czym znów obowiązuje przypisanie standardowe ustalone na przykład w czasie generacji systemu.

Wprowadzanie i wyprowadzanie danych

W skład systemu DOS/JS wchodzi system sterowania operacjami wejścia/wyjścia (IOCS), który steruje wprowadzaniem i wyprowadzaniem danych. Programy problemowe mogą wykorzystywać IOCS na dwóch poziomach: fizycznym (PIOCS), logicznym (LIOCS). Poziom PIOCS składa się z procedur, które znajdują się w programie nadzorczym i mają bezpośredni kontakt z urządzeniami, tzn. tylko PIOCS może inicjować transmisję i badać stan pracy kanałów i urządzeń.

Poziom LIOCS składa się z procedur wchodzących w skład programu problemowego oraz z procedur wymiennych (tzw. logicznych procedur wymiennych, na przykład otwierających zbiory). PIOCS wykorzystywany jest przez wszystkie programy, które korzystają z urządzeń zewnętrznych, niezależnie od tego, czy używają poziomu LIOCS, czy nie.

PIOCS nadzoruje współpracę systemu z kanałami, organizuje kolejki kanałowe, do których wpisywane są żądania rozpoczęcia operacji wejścia/wyjścia, inicjuje wykonywanie programów kanałowych oraz kontroluje ich przebieg. W przypadku niepoprawnie zakończonej transmisji — analizuje wyniki błędów.

LIOCS zajmuje się logiczną stroną wprowadzania i wyprowadzania danych: zapisuje i bada etykiety, wydziela rekordy logiczne z bloków, sprawdza poprawność budowy zbiorów, umożliwia stosowanie buforowania, obsługuje warunki „końca zbioru” i „końca nośnika”. LIOCS pozwala na tworzenie i przetwarzanie zbiorów o organizacji sekwencyjnej, indeskowo-sekwencyjnej i losowej.

W zbiorach sekwencyjnych kolejność rekordów wyznaczana jest automatycznie przez ich fizyczne miejsce w zbiorze. W zbiorach indeksowo-sekwencyjnych położenie rekordu w biorze określone jest przez klucz. W zbiorach losowych kolejność rekordów wyznaczona jest przez tak zwany algorytm adresacji, przyporządkowujący rekordowi miejsce w zbiorze.

Zbiory indeksowo-sekwencyjne i losowe mogą być pamiętane tylko na dyskach. Dla takich zbiorów istnieje możliwość zastosowania bezpośredniego dostępu do informacji, zmniejszającego czas przetwarzania.

Korzystanie z poziomu LIOCS ułatwia system makroinstrukcji. Użytkownik musi jedynie określić typ zbioru i sposób jego przetwarzania oraz podać, jakiego typu operacje chce wykonać. W momencie przetwarzania programu przez translator i program łączący, do programu dołączone zostaną odpowiednie moduły realizujące żądane czynności związane z wprowadzaniem/wyprowadzaniem danych.

Programy przetwarzające

Języki programowania

DOS/JS wyposażony jest w następujące języki programowania: Assembler, FORTRAN, PL/1 i RPG.

Język Assembler jest podstawowym językiem programowania komputerów Jednolitego Systemu. Jest to język o zastosowaniu uniwersalnym ukierunkowany maszynowo. Podstawową zaletą Assemblera jest możliwość stosowania symbolicznego zapisu języka maszyny (to znaczy — kodu operacji, adresów argumentu itd.) oraz automatycznego wyznaczenia adresów rozkazów i danych. Dane mogą być definiowane „jawnie” lub wynikowo w postaci literalów występujących bezpośrednio, jako argumenty rozkazów symbolicznych. Assembler ma możliwość definiowania i stosowania makrorozkazów i to zarówno systemowych (jak na przykład makrorozkazów IOCS), jak i definiowanych przez użytkownika. Makrorozkazy automatyzują w dużym stopniu proces pisania programu, zwłaszcza gdy w programie często powtarzają się ciągi złożone z tych samych rozkazów.

Używając makrorozkazów programista powoduje wygenerowanie (podczas tłumaczenia programu źródłowego) zamiast pojedynczego zdania — ciągu zdań. Bardziej złożone makrorozkazy o dużym stopniu elastyczności, mogą być wytwarzane przy stosowaniu rozkazów kompilacji warunkowej. Rozkazy kompilacji warunkowej pozwalają na uzależnienie procesu tłumaczenia programu od pewnych, ustalonych przez programistę, warunków. Stosując rozkazy kompilacji warunkowej programista może na przykład zmieniać typ rozkazu maszyny w zależności od typu danej.

Assembler umożliwia korzystanie z zapisów umieszczonych w bibliotece, tzn. pozwala programiście na użycie fragmentów programu, jak również makrodefinicji wcześniej skatalogowanych w bibliotece, bez konieczności ponownego ich pisania. Translator języka Assembler — zwany krótko Assemblerem — wymaga co najmniej 14K bajtów pamięci oraz zbiorów roboczych znajdujących się na dyskach lub na taśmach magnetycznych.

Język FORTRAN jest jednym z najszerzej stosowanych języków programowania, szczególnie nadającym się do rozwiązywania problemów matematycznych i technicznych. Jest on (w systemie DOS/JS) wersją języka FORTRAN IV zwaną BASIC FORTRAN IV (FORTRAN podstawowy).

Cechą charakterystyczną tego języka jest łatwość opisu wyrażeń matematycznych oraz budowy podprogramów. FORTRAN ma rozbudowany system wejścia/wyjścia. W jego skład wchodzi też podstawowe funkcje matematyczne takie, jak: sinus, cosinus, logarytm, funkcja wykładnicza itd. Obliczenia mogą być przeprowadzane na liczbach zmiennoprzecinkowych z podwójną precyzją, co pozwala na uzyskiwanie dokładniejszych wyników. Bogata jest

również biblioteka podprogramów, które mogą być wykorzystane w każdym programie napisanym w FORTRANIE.

Język PL/1 jest uniwersalnym, problemowo-zorientowanym językiem programowania nadającym się do zaprogramowania zarówno problemów numerycznych, jak i nienumerycznych. Zawiera on niezbędne środki programowania dla takich zagadnień, jak problemy ekonomiczne, techniczno-naukowe, czy przetwarzanie dowolnej informacji. Ze względu na łatwość przedstawiania algorytmów w tym języku, nadaje się on szczególnie do rozwiązywania zadań skomplikowanych i uniwersalnych (na przykład zintegrowanych systemów przetwarzania danych). Przy tworzeniu koncepcji języka PL/1 uwzględniono najlepsze rozwiązania występujące w już opracowanych i sprawdzonych w praktyce językach: FORTRAN-IV, ALGOL-60, COBOL, LISP.

Łatwość korzystania z języka PL/1 ułatwia fakt, że programista kodując algorytm z danej dziedziny przetwarzania informacji — nie musi znać całego języka, a tylko podzbiór zawierający te środki programowania, które są niezbędne do zakodowania danego algorytmu.

W systemie DOS/JS translator języka PL/1 nie realizuje pełnego języka, a tylko podzbiór dostosowany do urządzeń w jakie wyposażone będą komputery Jednolitego Systemu oraz do możliwości samego systemu. Podzbiór ten jest podzbiorem zamkniętym i efektywnie pracującym pod nadzorem DOS.

PL/1 nie jest językiem maszynowo-zorientowanym, ale posiada elementy, których efektywność bliska jest pracy na „poziomie maszyny”. W PL/1 istnieje możliwość deklaracji i korzystania z danych arytmetycznych (różnego rodzaju i z różną precyzją) i danych tekstowych (bitowych i znakowych). Operacje mogą odbywać się na danych różnych typów przy czym konwersja danych jest w dużym stopniu zautomatyzowana. PL/1 dopuszcza również stosowanie struktur danych. Wprowadzanie i wyprowadzanie danych napisanych w PL/1 ułatwia bogaty aparat wejścia/wyjścia.

Język RPG, jako problemowo-zorientowany język programowania, nadaje się szczególnie do rozwiązywania zagadnień, w których występują duże ilości danych do przetwarzania, nieskomplikowane obliczenia i skomplikowane wydruki. RPG umożliwia wykonanie tylko prostych operacji rachunkowych. Pozwala na wykonanie następujących funkcji:

- wprowadzanie i wyprowadzanie zbiorów z/na różne nośniki
- uzyskiwanie wszelkiego rodzaju sprawozdań, wykazów statystycznych i wydruków na drukarce wierszowej
- tworzenie i przetwarzanie zbiorów o różnych organizacjach
- wykonywanie prostych operacji arytmetycznych
- wyszukiwanie odpowiednich informacji ze zbiorów
- tworzenie zbiorów taśmowych z informacjami przygotowanymi do wydruku.

W programach napisanych w RPG można korzystać z podprogramów Assemblera.

RPG nadaje się przede wszystkim do zaprogramowania prostych zagadnień ekonomicznych:

- a) w rozliczaniu działalności przedsiębiorstw do ewidencji środków trwałych, obliczania plac, gospodarki materiałowej, rachunku kosztów, prowadzenia rozrachunków bieżących, bilansowania i rozliczania siły roboczej
- b) w bankach i kasach oszczędności
- c) w handlu do obliczania obrotów i ewidencji klientów.

Programowanie w RPG polega na wypełnieniu różnych, mających stałe pozycje, formularzy. Istnieją formularze do opisu zbiorów, nośników, rekordów wejściowych, operacji rachunkowych i rekordów wyjściowych.

RPG umożliwia przetwarzanie danych alfanumerycznych i liczb dziesiętnych. Może on przetwarzać zbiory o dowolnej organizacji lecz tworzy zbiory tylko

o organizacji sekwencyjnej lub indeksowo-sekwencyjnej.

„Technika formularzowa”, zastosowana dla RPG, umożliwia korzystanie z tego języka bez posiadania nawet elementarnych wiadomości z dziedziny programowania. Programy oznaczają się dużą przejrzystością i są łatwe do sprawdzania.

Translator RPG tworzy (generuje) — zgodnie z podanymi informacjami — program złożony z pewnej liczby procedur, do których mogą być dołączone podprogramy użytkownika. Tak więc szkielet utworzonego programu jest stosunkowo „sztuczny”. Programista ma jednak możliwość sterowania przebiegiem programu za pomocą specjalnych wskaźników.

Translatory wszystkich języków sprawdzają syntaktyczną poprawność programów źródłowych i informują o zaistniałych błędach, podając, w którym miejscu programu i z jakich przyczyn one wystąpiły.

Programy usługowe wykonują standardowe czynności, które są niezbędne, niezależnie od problemu, jaki chcemy rozwiązać. Do programów usługowych należą: program łączący, bibliotekarz, systemowe programy pomocnicze, Autotest i programy sortowania zbiorów.

Program łączący. Każdy z programów, po przetłumaczeniu przez translator musi być zredagowany przez program łączący. Program łączący, zgodnie z podanymi mu zdaniami sterującymi, buduje — z modułów wynikowych — fazy programu gotowego do wykonania. Każdej z faz programu nadaje się adres pamięci operacyjnej, pod który ma ona być wprowadzona. Dodatkowo można określić punkt wejścia do fazy, to znaczy podać adres, od którego należy rozpocząć jej wykonywanie. Program łączący może połączyć w jeden program moduły wynikowe otrzymane w rezultacie przetłumaczenia różnych programów źródłowych napisanych w różnych językach programowania i w różnym czasie. Program łączący może współpracować z biblioteką modułów i uzupełnić redagowany program o brakujące części np. o podprogramy znajdujące się w bibliotece. Po zredagowaniu program może być skatalogowany na stałe do biblioteki faz, co umożliwia wielokrotne jego użycie. Funkcje i powiązania programu łączącego w poszczególnych etapach przetwarzania programów ilustruje rys. 2.

Bibliotekarz. W systemie DOS/JS rozróżnia się biblioteki trzech rodzajów:

- bibliotekę faz CL (konieczna)
- bibliotekę modułów (RL)
- bibliotekę tekstów (SL).

Do każdej biblioteki dołączony jest katalog opisujący poszczególne pozycje znajdujące się w bibliotece.

Biblioteka CL zawiera programy składające się z jednej lub kilku faz (faza może być pełnym programem, albo może wchodzić w skład programu wielofazowego). Programy w CL mogą być wprowadzone do pamięci operacyjnej i wykonane pod kontrolą systemu DOS/JS.

Biblioteka RL zawiera moduły przetłumaczone przez translatory języków programowania. Moduły te mogą być łączone z innymi modułami bez potrzeby ponownej kompilacji. Wykonuje to program łączący.

Biblioteka SL zawiera książki, z których każda zawiera ciąg zdań źródłowych lub makrodefinicję. Książka może być włączona do programu źródłowego podczas procesu kompilacji (np. przez zdanie COPY w Assemblerze). Zdania pamiętane w bibliotece SL występują w postaci zwartej, tzn. pominięte są wszystkie spacje. Podczas włączania książki do modułu źródłowego zdaniom z książki przywraca się postać pierwotną.

Bibliotekarz umożliwia konserwację bibliotek, wprowadzanie informacji do bibliotek, wyprowadzanie informacji z bibliotek i kopiowanie bibliotek. Składa się on z programów MAINT, CSERV, SSERV, DSERV, CORGZ.

Program MAINT, wykonujący konserwację bibliotek, umożliwia:

- dopisanie do bibliotek nowych elementów
- skasowanie z bibliotek istniejących elementów
- przemianowanie elementów znajdujących się w bibliotece
- kondensację bibliotek
- przemieszczanie bibliotek systemowych
- aktualizację bibliotek SL.

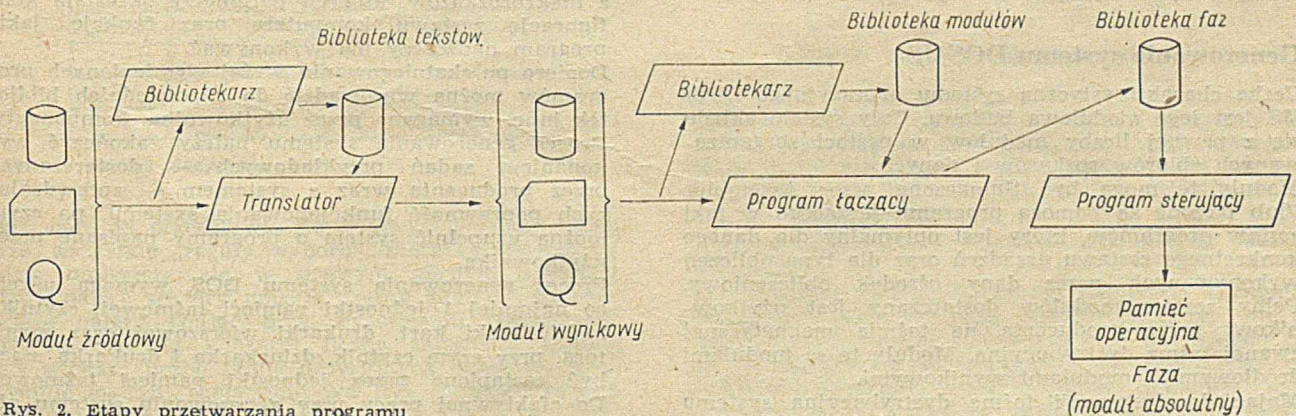
Programy CSERV, RSERV i SSERV umożliwiają wydrukowanie i/lub wydziurkowanie elementów z odpowiednich bibliotek.

Drukowanie katalogów umożliwia program DSERV. Za pomocą programu CORGZ można wykonać następujące czynności:

- utworzyć biblioteki prywatne użytkownika
- przepisać z jednej biblioteki do drugiej wybrane fazy, moduły lub książki.

Systemowe programy pomocnicze. Systemowe programy pomocnicze zapewniają wykonanie następujących czynności organizacyjnych dla systemu DOS/JS:

- inicjowanie pakietów dysków magnetycznych
- wydzielanie ścieżek rezerwowych na dyskach
- kasowanie informacji z dysków przez zapis sztucznych rekordów
- wypisywanie tablicy VTOC (informacje o zawartości dysków).



Rys. 2. Etapy przetwarzania programu

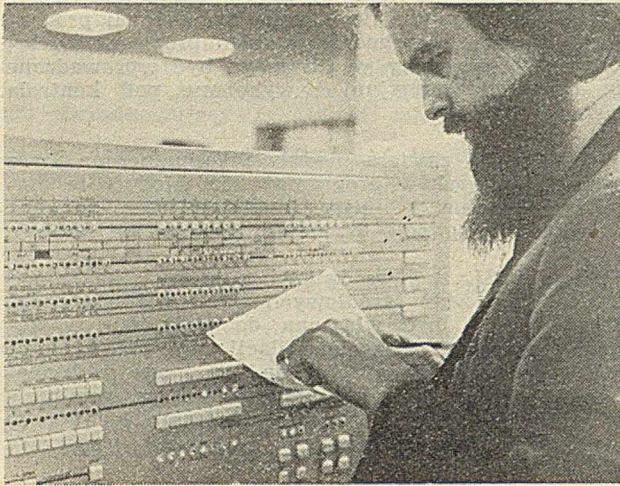


Foto 3. Inż. W. Jakacki przy pulpicie R-30 w ELWRO

Za pomocą systemowych programów pomocniczych można również przepisywać informacje z jednego nośnika na drugi, tworzyć duplikaty zbiorów i przechowywać system na taśmie magnetycznej. Zbiory mogą być kopiowane z zachowaniem określonych przez użytkownika reguł odnośnie ich postaci.

Programy sortowania. DOS/JS zawiera: program sortowania zbiorów taśmowych oraz program sortowania zarówno zbiorów taśmowych, jak i dyskowych. Programy sortowania umożliwiają posortowanie zbiorów³ w kolejności rosnącej, jak i malejącej, zgodnie z kluczem składającym się maksymalnie z 12 pól. Obydwa programy przetwarzają etykiety zbiorów, umożliwiają stosowanie zbiorów wieloszpulowych i/lub wielopakietowych, pozwalają na włączenie procedur użytkownika, zawierają procedury przerywające i wznowiające operacje sortowania.

Program sortowania zbiorów taśmowych może posortować jednocześnie 8 zbiorów i połączyć 7 zbiorów w jeden zbiór posortowany.

Program sortowania zbiorów taśmowych i dyskowych może posortować jednocześnie 9 zbiorów oraz połączyć 8 zbiorów w jeden zbiór posortowany. Program ten można włączyć, jako podprogram, do programu użytkownika.

Program samotestujący (Autotest). Program Autotest zwiera środki testujące programy, przy czym jest on szczególnie przydatny dla testowania programów napisanych w języku Assembler. Umożliwia on wykonywanie następujących funkcji:

- automatycznej korekty, przy czym rozkazy mogą być zamienione, dodane lub usunięte, bez konieczności ponownej kompilacji programu
- wypisania zawartości pamięci operacyjnej; podczas pracy programu można uzyskać wydruk zawartości określonego obszaru pamięci
- wypisania zawartości rejestrów i specjalnych informacji umieszczonych w bajtach 24—127.

Generowanie systemu DOS/JS

Cechą charakterystyczną systemu operacyjnego DOS/JS jest jego modułowa budowa. Cały system składa się z pewnej liczby modułów w postaci zorganizowanych zbiorów rozkazów i danych.

Moduły te mogą być tłumaczone przez Assembler i/lub łączone za pomocą programu łączącego w taki zestaw programów, który jest optymalny dla danego konkretnego zestawu urządzeń oraz dla typu obliczeń wykonywanych przez dany ośrodek obliczeniowy. Pełny zestaw modułów dostarczany jest użytkownikowi przez producenta na taśmie magnetycznej zwanej taśmą dystrybucyjną. Moduły te są modułami źródłowymi i modułami wynikowymi.

Mając do dyspozycji taśmę dystrybucyjną systemu DOS/JS, użytkownik musi wykonać ściśle określone

czynności w celu uzyskania systemu zdolnego do pracy w danych warunkach.

Pod pojęciem systemu zdolnego do pracy należy rozumieć jeden lub więcej pakietów dyskowych zawierających w odpowiednich bibliotekach wszystkie (niezbędne do wykonywania obliczeń żadanego typu) komponenty systemu DOS, jak również zbiory robocze wymagane przez te komponenty. Proces tworzenia systemu zdolnego do pracy nazywa się generowaniem systemu, które odbywa się pod kontrolą tzw. prasystemu, znajdującego się na taśmie dystrybucyjnej.

Pierwszym krokiem procesu generowania systemu jest zainicjowanie pakietów dyskowych, które będą wykorzystywane w dalszych etapach generowania. Następnie odtwarza się zawartości zbiorów „CL/SL” i „CL/RL” znajdujących się na taśmie dystrybucyjnej, przepisując je na dwa pakiety dyskowe. W ten sposób otrzymuje się dwa systemy: jeden z nich zawiera bibliotekę faz i bibliotek tekstów źródłowych, drugi zaś bibliotekę faz i bibliotekę modułów. Zawartości obu bibliotek faz są sobie równe, składają się na nie komponenty prasystemu. Powyższe czynności wykonuje się przy pomocy specjalnych programów znajdujących się na początku taśmy dystrybucyjnej (programu inicjowania i programu odtwarzania).

Kolejnym, ważnym etapem generowania jest ustalenie wielkości i zawartości bibliotek tworzonego systemu. Niezależnie od wyposażenia, jakim dysponuje użytkownik, jak również od typu obliczeń przez niego wykonywanych, biblioteka faz musi zawierać programy sterujące, program łączący i programy bibliotekarza, jak również pewną grupę procedur wymiennych związanych z pracą programów sterujących. Pozostałe komponenty mogą być włączone do systemu lub nie, w zależności od życzeń użytkownika; przy czym należy zwrócić uwagę, że włączenie do systemu komponentu opcjonalnego może spowodować konieczność włączenia dalszych modułów opcjonalnych (na przykład: wraz z translatorem języka programowania należy włączyć odpowiednie moduły wejścia/wyjścia związane z tym translatorem). Zawartość bibliotek tworzy się, przenosząc i przetwarzając odpowiednie moduły z bibliotek SL i RL prasystemu.

Ze względu na to, że dobrze zaplanowane i prawidłowo rozlokowane biblioteki mają poważny wpływ na efektywną pracę systemu — czynności związane z ustaleniem wielkości i zawartości bibliotek należy przeprowadzić szczególnie dokładnie.

Przy planowaniu wielkości bibliotek należy zwrócić uwagę, na fakt, że ponieważ w bibliotece faz zapamiętywane są programy wykonywane, to właśnie ta biblioteka może ulec największemu rozszerzeniu w przyszłości.

Pierwszymi komponentami wprowadzanymi do bibliotek tworzonego systemu muszą być programy sterujące, program łączący i programy bibliotekarza. Szczególną uwagę należy zwrócić na wygenerowanie odpowiedniego programu nadzorczego. Generowanie tego programu polega na przetłumaczeniu a następnie skatalogowaniu w bibliotece faz programu napisanego w języku Assembler. Program ten składa się z makrorozkazów, których parametry określają konfigurację zestawu komputera oraz funkcje, jakie program nadzorczy ma wykonywać.

Dopiero po skatalogowaniu wyżej wymienionych programów można wprowadzić do odpowiednich bibliotek inne, wymagane przez użytkownika, komponenty. Proces generowania systemu należy zakończyć wykonaniem zadań przykładowych — dostarczanych przez producenta wraz z systemem — sprawdzających poprawność funkcjonowania systemu, po czym można uzupełnić system o programy napisane przez użytkownika.

Proces generowania systemu DOS wymaga użycia co najmniej 1 jednostki pamięci taśmowej, czytnika i dziurkarki kart, drukarki wierszowej oraz monitora, przy czym czytnik, dziurkarka i drukarka mogą być zastąpione przez jednostki pamięci taśmowej. Do efektywnej pracy przy generowaniu systemu nie wystarcza zwykle jedna jednostka pamięci dyskowej.

Mgr inż. BENEDYKT BURGIELSKI ukończył studia na Wydziale Elektroniki Politechniki Wrocławskiej w Instytucie Cybernetyki Technicznej (1969). Od ukończenia studiów pracuje w Instytucie Konstrukcji i Eksploatacji Maszyn Politechniki Wrocławskiej. Zajmuje się teorią projektowania inżynierskiego oraz implementacją procedur projektowych wkomputerowanym systemie Politechniki Wrocławskiej WASC/ADID.



BENEDYKT BURGIELSKI

65.012.122.681.322.06:62-52.001:13

Politechnika Wroclawska

Wykorzystanie tablic decyzyjnych w programowaniu oraz rozwiązywaniu problemów typu projektowego

Przedstawiono zasady budowy tablic decyzyjnych jako zorientowanego problemowo języka programowania wyższego szczebla. Scharakteryzowano algorytm przekształcający tablicę decyzyjną w strukturę dendrytową, dającą się łatwo zaprogramować za pomocą dowolnego języka podstawowego. Podano przykład wykorzystania tablic decyzyjnych do zautomatyzowanego projektowania suwnic z pomocą komputera IBM 360/40.

Rodzaje tablic decyzyjnych

W praktyce występują trzy rodzaje tablic decyzyjnych: prosta, uogólniona i łączna.

Rys. 2 przedstawia przykład prostej tablicy decyzyjnej

- „T” — oznacza „tak”, tzn., że *i*-ty warunek jest prawdziwy dla *k*-tej reguły
- „N” — oznacza „nie”, tzn., że dany warunek nie jest prawdziwy dla reguły, przy której występuje duża litera „N”
- „—” — oznacza „pusty” lub „obojętny”, tzn., że warunek ten jest obojętny lub nie ma zastosowania. W sektorze akcji znak „—” oznacza, że nie zostaje podjęta żadna akcja przy wystąpieniu konkretnej konfiguracji warunków. Logicznie „—” oznacza, że warunek może być prawdziwy lub fałszywy
- „X” — oznacza podjęcie danej akcji.

Z prostą tablicą decyzyjną mamy do czynienia wtedy, gdy określonym warunkom są przyporządkowane ściśle określone reguły wyboru.

Tablice decyzyjne stanowią, dzięki swojej formalnej budowie pełny system kombinacji warunków. W praktyce z teoretycznie możliwych reguł wyboru, wystarczająca jest tylko ich część, pozwalająca uchwycić wszystkie rzeczywiście występujące przypadki.

	R1	R2	R	INNE
A=B	—	T	N	—
C=D	T	N	—	—
E=F	N	—	T	—
X	X	—	X	—
Z	—	X	X	X

Rys. 2. Prosta tablica decyzyjna

Morfologie tablic decyzyjnych

Tablice decyzyjne są tabelaryczną formą przedstawiania logiki podejmowania decyzji. Metoda tablic decyzyjnych znalazła zastosowanie w procesie automatyzacji prac projektowych dotyczących problemów powtarzalnych i typowych. Tablice decyzyjne spełniają w procesie automatyzacji prac projektowych podwójną rolę: umożliwiają one przejrzyste przedstawienie zależności technicznych występujących w procesie projektowania, a równocześnie mogą być zrozumiałe dla komputera jako program. Spośród wielu korzyści wynikających ze stosowania tej metody należy wymienić przede wszystkim:

- znaczne zredukowanie zapotrzebowania pojemności pamięci komputera
- łatwość programowania
- czytelność dokumentacji.

Budowa tablic decyzyjnych

W tablicach decyzyjnych przedstawia się związki przyczynowo-skutkowe, które można zapisać w postaci zdań oznajmująco-warunkowych typu „Jeżeli — To”, np.: jeżeli występują określone warunki, to należy podjąć określone akcje. Formalnie tablica decyzyjna podzielona jest na cztery sektory. Wszyst-

Warunki	Reguły wyboru			
	Reguła 1	Reguła 2	Reguła 3	INNE
„Jeżeli”				
„To”				

Rys. 1. Tablica decyzyjna

kie warunki przedstawia się w lewym górnym sektorze, wszystkie „reguły wyboru”, które należy sprawdzić umieszcza się w prawym górnym sektorze. Dwa dalsze sektory opisują szereg akcji — działań, które należy podjąć przy wystąpieniu określonych warunków. Dla większej liczby zależności stosuje się większą liczbę tablic decyzyjnych, które są wzajemnie logicznie połączone. Na Rys. 1 przedstawiono ogólny układ tablicy decyzyjnej.

Całkowitość tablic decyzyjnych można w tym przypadku zapewnić za pomocą reguły zawierającej w sobie wszystkie kombinacje warunków nie ujętych w poprzednich regułach. Taką regułą wyboru nazywa się regułą INNE.

Rys. 3 przedstawia przykład uogólnionej tablicy decyzyjnej.

Z uogólnioną tablicą decyzyjną mamy do czynienia wtedy, gdy dany warunek jest wieloznaczny, np. A może równać się zarówno B jak i C. Ze względu na to, że algorytmy tłumaczące tabele decyzyjne na podstawowy język programowania dokonują tego przekładu tylko z prostych tablic decyzyjnych, należy

przejsć z uogólnionej tablicy decyzyjnej na tablicę decyzyjną prostą. Przykład takiego przejścia pokazano na Rys. 4 i 5.

	R1	R2	R3	INNE
A=	-	B	C	-
C=D	T	N	-	-
E=	F	-	F	-
X	X	-	X	-
Y	-	X	X	X

Rys. 3. Uogólniona tablica decyzyjna

Tablica decyzyjna z rys. 4 jest równoważna prostej tablicy decyzyjnej z rys. 5. Tablica z rys. 5 może być natomiast uproszczona do postaci z rys. 6.

A	=B	>C	≠B
M,N	≥	<	-
X<	Z	-	W

Rys. 4. Uogólniona tablica decyzyjna

Rys. 5. Prosta tablica decyzyjna (z nadmiarem warunków)

	R1	R2	R3	INNE
A=B	T	-	-	-
A>C	-	T	-	-
A≠B	-	-	T	-
M≥N	T	-	-	-
M<N	-	T	-	-
X<Z	T	-	-	-
X>W	-	-	T	-

Rys. 6. Uproszczona tablica decyzyjna z rys. 5

	R1	R2	R3	INNE
A=B	T	-	N	-
A>C	-	T	-	-
M≥N	T	N	-	-
X<Y	T	-	-	-
X<W	-	-	T	-

Rys. 7. Łączna tablica decyzyjna

Łączną tablicę decyzyjną przedstawiono na rys. 7.

Błędy przy układaniu tablic decyzyjnych

Przy układaniu tablic decyzyjnych występuje możliwość popełniania błędów. Błędami tymi mogą być nadmiar oraz sprzeczność warunków.

	R1	R2
W1	-	N
W2	T	T

Rys. 8. Tablica wzorcowa dla przykładów wykrycia błędów

Jeżeli na każdym poziomie warunków, dwie lub więcej reguł wyboru nie zawierają co najmniej jednej pary T,N, w każdym szeregu, to mamy do czynienia z nadmiarem lub sprzecznością.

Jeżeli akcje dla reguł 1 i 2 z tablicy na rys. 8 są te same, to występuje nadmiar warunków (rys. 9), natomiast jeżeli są one różne, to warunki wzajemnie sobie zaprzeczają (rys. 10).

	R1	R2
W1	-	N
W2	T	T
A1	X	-
A2	-	X

Rys. 9. Nadmiar

	R1	R2
W1	-	N
W2	T	T
A1	X	X
A2	-	-

Rys. 10. Sprzeczność

Dalsze informacje na temat istoty oraz stosowania tablic decyzyjnych Czytelnik znaleźć może w artykule J. Niedźwiedzkiego „Tablice decyzji — Struktura i zastosowanie” opublikowanego w nr 1/1972 INFORMATYKI.

Tablice decyzyjne jako język programowania wyższego szczebla zorientowany problemowo

Ogólna charakterystyka problemu

Tablice decyzyjne można, za pomocą formalnych instrukcji, przekształcać na program nadający się do bezpośredniej realizacji na maszynie cyfrowej. Może to być dokonane przez programistę lub przez translator. Obecnie istnieje wiele translatorów tablic decyzyjnych na podstawowe języki programowania. Są to przykładowo DETAB/65 i Agenta dla języka COBOL lub DLT (Decision Logic Translator) — dla języka FORTRAN. Poniżej zostanie scharakteryzowany algorytm przekształcający tablicę decyzyjną w strukturę dendrytową, dającą się w łatwy sposób zaprogramować za pomocą dowolnego języka podstawowego. Algorytm ten minimalizuje liczbę porównywalnych elementów w programie, a zatem minimalizuje zapotrzebowanie pojemności wykorzystywanej pamięci operacyjnej. Przed przystąpieniem do przekształcenia tablicy decyzyjnej w strukturę dendrytową należy zredukować do minimum liczbę wykonywanych niezależnych akcji i reguł wyboru. W podanym opisie algorytmu pominięto dla wygody sektor akcji. Należy więc zaznaczyć, że każda z pokazanych na schematach blokowych „reguł wyboru”, może być reprezentowana przez jedną lub więcej akcji specyficznych dla każdej z tych reguł.

Liczba kolumnowa 4 4 2

	R1	R2	R3	Liczba kreskowna
W1	N	-	T	4
W2	N	T	-	2
W3	-	T	N	4
W4	-	-	N	8

Rys. 11. Określenie parametrów tablicy decyzyjnej

Przykład określenia tych szczególnych parametrów tablic decyzyjnych podano na rys. 11.

Algorytm ten polega na tym, że dokonuje się wyboru i wydziela określony warunek z sektora warunków. Warunek ten dzieli pierwotną tablicę decyzyjną na dwie podtablice w zależności od tego, czy były spełnione reguły wyboru dla wybranego warunku. Warunek ten staje się pierwszym elementem porównującym na schemacie blokowym. Proces ten

prowadzi się dopóty, dopóki każda reguła nie stanie się jedyną dla jednego warunku lub dopóki nie okaże się, że tablica pierwotna zawiera nadmiar albo jest sprzeczna.

Algorytm minimalizujący zapotrzebowanie pojemności pamięci

Krok 1. Sprawdzenie tablicy decyzyjnej na nadmiar lub sprzeczność w sposób omówiony powyżej.

Krok 2. Określenie tych wierszy, które mają najmniejszą „liczbę kreskową”. Liczba ta jest miarą obojętności gdyż kreska oznacza, że dany warunek może być albo prawdziwy, albo fałszywy, a więc nie ma to wpływu na podjętą decyzję.

Definicje.

Liczbę kolumnową k_i wyznacza się dla każdej kolumny tablicy decyzyjnej i określa się ją, jako wyrażenie $k_i = 3^{r_i}$, gdzie r_i oznacza liczbę kresek w i -tej kolumnie tablicy decyzyjnej. ($i = 1, 2, 3, \dots$) Jeżeli $r_i = 0$, to liczba kolumnowa dla i -tej kolumny wynosi 1. Liczbę kreskową K_j wyznacza się dla każdego wiersza tablicy decyzyjnej. Liczba kreskowa dla danego wiersza w_j jest sumą liczb kolumnowych dla tych kolumn, dla których w danym wierszu w_j wystąpi kreska.

Przykładowo: liczba kreskowa K_1 (dla pierwszego wiersza) na rys. 11 wynosi 4, ponieważ w pierwszym wierszu tej tablicy decyzyjnej wystąpiła kreska tylko w drugiej kolumnie, dla której liczba kolumnowa k_2 wynosi 4. Przykład określenia tych szczególnych parametrów tablic decyzyjnych podano na rys. 11.

W tym przypadku wiersz drugi ma najmniejszą „liczbę kreskową” i od tego wiersza rozpoczyna się proces dekompozycji tej tablicy decyzyjnej.

Krok 3. Określenie parametru „delta”.

Jeżeli dwa lub więcej wierszy mają minimalną liczbę kreskową, to wyboru dokonuje się na podstawie parametru „delta”. Jest on określony przez absolutną wartość różnicy między sumami liczb kolumnowych dla elementów T i elementów N w każdym wierszu.

Liczba kolumnowa 1 2 1 2 2 2

	R1	R2	R3	R4	R5	R6	Liczba kreskowa	Delta	W_k
W1	T	N	N	N	N	-	2	6	
W2	T	N	T	-	T	N	2	0	
W3	N	N	N	T	-	T	2	0	
W4	N	-	N	N	T	T	2	0	

Rys. 12. Określenie parametru delta

Przykładowo, w trzecim wierszu tablicy pokazanej na rys. 12 parametr delta wynosi 0, ponieważ sumy liczb kolumnowych w tym wierszu dla elementów N i elementów T są jednakowe ($1+2+1=2+2$). W pokazanym przykładzie (rys. 12) wszystkie wiersze mają jednakową minimalną liczbę kreskową, zaś parametry „delta” są dla nich następujące:

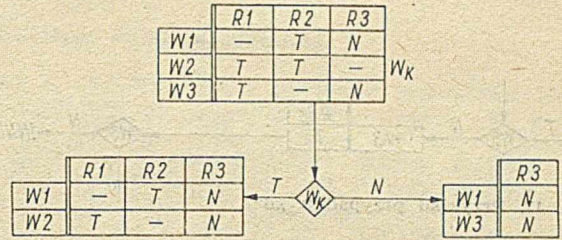
$$\begin{aligned} \Delta 1 &= 2+1+2+2-1=6 \\ \Delta 2 &= 1+1+2-2-2=0 \\ \Delta 3 &= 1+2+1-2-2=0 \\ \Delta 4 &= 1+1+2-2-2=0 \end{aligned}$$

Wybrany w ten sposób wiersz oznacza się indeksem k , w tym przykładzie jest to wiersz pierwszy. $W1 = W_k$.

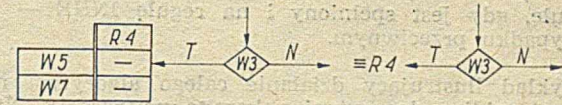
Krok 4. Rozdzielenie pierwotnej tablicy decyzyjnej na dwie podtablice.

Podtablice zawierają jedną lub więcej reguł wyboru i o jeden wiersz mniej, niż tablica pierwotna. Jak

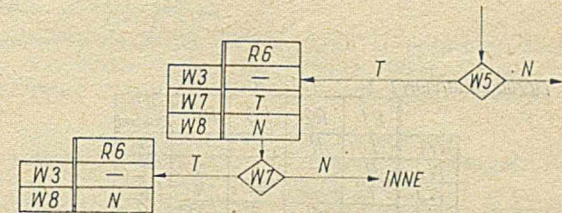
pokazano na Rys. 13 T — strzałka prowadzi do podtablicy zawierającej wszystkie reguły wyboru mające decyzję T w k -tym wierszu pierwotnej tablicy decyzyjnej.



Rys. 13. Określenie sposobu rozdzielania tablic decyzyjnych



Rys. 14. Przykład przypadku „a”



Rys. 15. Przykład przypadku „b”

Podtablica połączona z tablicą pierwotną wskaźnikiem N i k -tym warunkiem utworzona jest podobnie. Należy dodać, że obie podtablice zawierają też reguły wyboru, które były obojętne (miały kreski) dla k -tego wiersza tablicy pierwotnej.

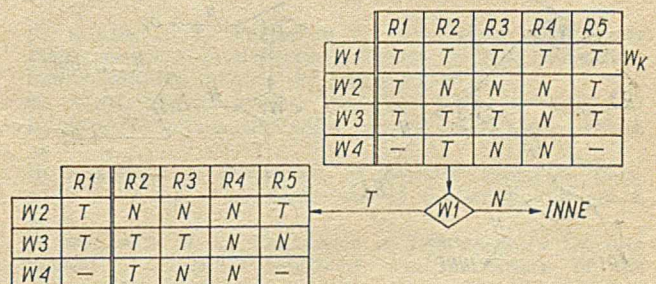
Krok 5. Powrót do kroku 1.

Jeżeli podtablica zawiera więcej, niż jedną regułę wyboru, należy wrócić do kroku 1.

Krok 6. Określenie końcowych elementów schematu blokowego.

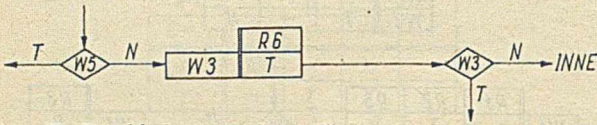
W kroku tym wystąpić mogą następujące warianty rozwiązań:

- jeżeli wskaźnik skierowany jest na podtablicę zawierającą dokładnie jedną regułę, to należy zastąpić podtablicę tą regułą. Przykład podano na rys. 14.
- jeżeli wskaźnik skierowany jest na podtablicę zawierającą jedną regułę i kreski nie dla wszystkich warunków, to należy wybrać dowolny wiersz nie zawierający kreski, jako wiersz wyróżniony W_k . Wybrany wiersz rozdzieli tę podtablicę na podtablicę zawierającą o jeden wiersz mniej i na element przeciwny, który określa się, jako reguła INNE.



Rys. 16. Przykład przypadku „c”

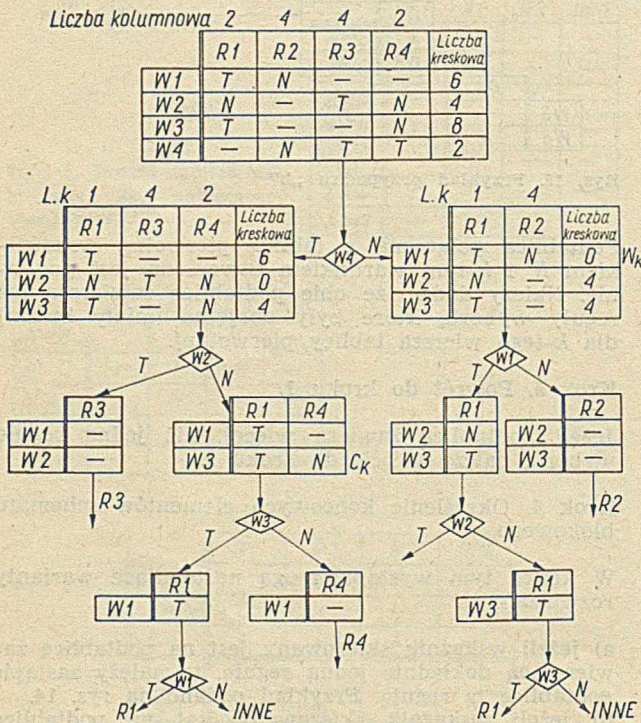
c) jeżeli wskaźnik nie prowadzi do podtablicy, to prowadzi do reguły INNE, jak pokazano na rys. 16.
 d) jeżeli wskaźnik prowadzi do podtablicy zawierającej tylko jedną regułę wyboru i reguła ta obejmuje tylko jeden warunek, którego wartość jest T



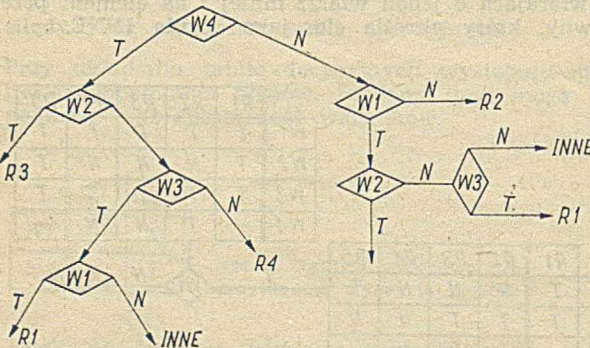
Rys. 17. Przykład przypadku „d”

lub N, to wtedy warunek ten rozgałęzia się na tę regułę, gdy jest spełniony i na regułę INNE — w przypadku przeciwnym.

Przykład ilustrujący działanie całego algorytmu na jednej tablicy decyzyjnej pokazano na rys. 18. Należy zaznaczyć, że reguły nie wpisane do tej tablicy należy uważać za reguły INNE.



Rys. 18. Przykład działania całego algorytmu



Rys. 19. Schemat blokowy dla tablicy decyzyjnej z rys. 18 (struktura dendrytowa)

Obok algorytmów minimalizujących pojemność wykorzystywanej pamięci, istnieją algorytmy minimalizujące czas przebiegu programu. W zależności od rodzaju głównych ograniczeń systemowych programista może skorzystać z jednego z tych dwóch algorytmów. Zbudowanie tablicy decyzyjnej jest jednorazowym procesem, który odpowiada programowaniu. Język tablic decyzyjnych jest jednak tak zorientowany problemowo, że mogą być one wykonywane i wykorzystywane przez fachowca technika, bez pomocy programisty. Tym samym odpowiedzialność za prawidłowe ułożenie programu spada bezpośrednio na projektanta. Metoda tablic decyzyjnych, jako forma języka programowania wyższego szczebla, stanowi więc dogodny pomost pomiędzy systemem komputerowym, a projektantem urządzeń.

Wykorzystanie tablic decyzyjnych do projektowania urządzeń technicznych

Wykorzystanie tablic decyzyjnych w projektowaniu urządzeń ilustruje przykład systemu automatycznego projektowania suwnic w firmie DEMAG (RFN). Przejście tego przedsiębiorstwa na zautomatyzowane systemy sterowania produkcją i planowania oraz na automatyczne, wielowariantowe projektowanie wszystkich wyrobów zostało zrealizowane na komputerze IBM 360/40 z pamięcią operacyjną o pojemności 128K oraz pamięcią dyskową IBM 2314. Logika konstrukcji i technologii wyrobów zaprogramowana została przy użyciu metody tablic decyzyjnych. Komunikacja tablic decyzyjnych z danymi zawartymi w banku danych konstrukcyjno-technologicznych (BOMP) umożliwia łatwą i szybką modyfikację zespołów standardowych.

Ręczne i maszynowe opracowanie zamówień

• Projektowanie

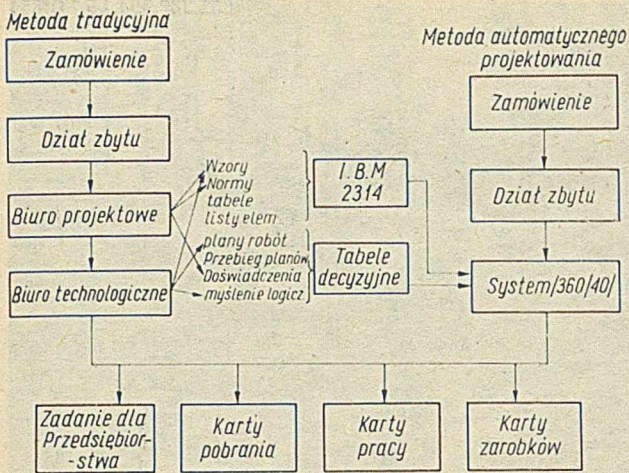
Różnorakie żądania klientów odnośnie budowy suwnicy standardowej wymagają obszernego opracowania technicznego. Biuro konstrukcyjne musi na tej podstawie wyznaczyć takie parametry suwnicy jak: rozpiętość, udźwig, wysokość podnoszenia, napięcie prądu głównego, napięcie prądu sterowania, prędkość jazdy, itp. Stąd na podstawie obliczeń i rysunków można wyznaczyć wymiary warsztatowe oraz określić montaż zespołów. Dla typowej suwnicy trzeba ustalić np. do 150 wymiarów warsztatowych oraz do 200 pozycji odbierania elementów ze składu.

• Opracowanie technologiczne

Technolodzy muszą na podstawie danych z biura projektowego ustalić kolejność prac, środki ich realizacji oraz skalkulować czas wytwarzania. Podstawą tego procesu przy dotychczasowym ręcznym sposobie opracowania było ok. 30 000 planów pracy. Rys. 20 pokazuje drogę zamówienia — przy metodzie tradycyjnej i metodzie opracowania automatycznego — od działu zbytu poprzez biuro projektowe, biuro technologiczne aż do przekazania zakładowi produkcyjnemu całości dokumentacji w formie zadania produkcyjnego, kwitów pobrania materiałów, kart pracy oraz kart zarobków.

Aby wykonanie tego procesu mogła przejąć maszyna cyfrowa należało rozwiązać kilka problemów. Przede wszystkim podane przez klienta wymagania dotyczące wyrobu trzeba było sformułować w sposób dostosowany do wymagań systemu. Należało też zebrać wszystkie reguły, wg których komputer może symulować wszystkie prace w biurze konstrukcyjnym i technologicznym. Zostało to wykonane za pomocą tablic decyzyjnych. Następnie formę i zawartość wytworzonej przez komputer dokumentacji trzeba było dostosować do wymagań przedsiębiorstwa oraz do dalszego, zintegrowanego przetwarzania.

Procedura postępowania przy rozwiązywaniu tego zagadnienia składa się z pakietu programów, który obejmuje programy wprowadzania i przetwarzania danych, sterowanie całością oraz wyprowadzanie wydawnictw wynikowych.



Rys. 20. Przejście z metody ręcznej na opracowanie maszynowe

Zorientowana problemowo główna część pakietu opisana jest za pomocą tablic decyzyjnych dla FORTRANU, natomiast operacje wejścia/wyjścia, współpraca z pamięcią w zakresie dostępu do specyfikacji części (BOMP) oraz sterowanie całością — opisane są w języku COBOL. Cały pakiet programów zajmuje w pamięci 56 000 bajtów.

Przykłady tablic decyzyjnych

Wybór blachy

Rys. 21 przedstawia tablicę decyzyjną, która pozwala dokonać wyboru blachy na dźwigar suwnicy.

Nr T.D	Nazwa tablicy: Wybór tablicy	Reguła 1	Reguła 2
1 2	Jeżeli ITRAG = 500	T	T
1 4	Jeżeli ISPAN =	15200	17000
1 6	Jeżeli ITRAG = 750	T	T
1 8	Jeżeli ISPAN <=	14300	17000
1 10	Wybierz blachę J.STEG	231310	228404

Rys. 21. Tablica decyzyjna „wybór blachy”

Numer T.D	Nazwa tablicy: Obliczanie czasu piaskowania	R1	R2
3 2		R1	R2
3 4	Jeżeli ISTE G =	231310	231310
3 6	Jeżeli MASSA =	6000	12000
3 8	Potóż GRUNW =	2.5	2.0
3 10	Potóż PROG =	2.0	2.0
3 12	Potóż $x = \text{MASSA}/1000$	X	X
3 14	Idź do tablicy 29	X	X
3 16	Potóż $\text{ITEO1} = Y$	X	X
3 18	Idź do tablicy 32	X	X
29 2	Obliczanie wg formuły $Y = A + B$		
29 4	Potóż $Y = \text{PROG} * x + \text{GRUNW}$	X	X
29 6	Idź do tablicy 32	X	X

Rys. 22. Tablica decyzyjna do obliczania czasu piaskowania

Jeżeli udźwig (ITRAG) = 500 kg, a rozpiętość wynosi (ISPAN) = 15 200 mm, lub udźwig = 750 kg, a rozpiętość 14 300 mm wtedy wybrana zostanie blacha o symbolu (ISTEG) 231310. Symbol ten oznacza blachę o odpowiednich właściwościach fizycznych i o konkretnych wymiarach. Dla innych warunków wybrana zostanie blacha o innym odpowiednim symbolu.

Obliczanie czasu piaskowania

Jest to już przykład z dziedziny zastosowań tablic decyzyjnych do przetwarzania danych w biurze technologicznym. Gdy symbol blachy (ISTEG) = 231310, a długość dźwigara wynosi (MASSA) = 6000 mm, to należy wstawić — jako wartość podstawową — (GRUNW) = 2,5 oraz progresję (PROG) = 2,0. W tablicy wylicza się czas piaskowania wg wzoru $Y = ax + b$. W tablicy 32 wartość tego czasu zostaje zaokrąglona wg odpowiednich zasad. Rozkazy „Idź do Tablicy 29 i 32” określają te tablice, jako podprogramy (tablice te nie zostały zamieszczone w niniejszym opracowaniu). Po wyliczeniu czasu piaskowania program wraca do następnej wskazówki. Czas pracy zostaje umieszczony na odpowiednich dokumentach wynikowych.

Korzyści ze stosowania metody tablic decyzyjnych

Zastosowanie tablic decyzyjnych w opisanym zakresie czynności przyniosło firmie DEMAG w ciągu krótkiego czasu eksploatacji następujące korzyści:

- skrócenie czasu wykonania zamówień
- całkowite wyeliminowanie kosztów związanych ze sporządzeniem harmonogramów prac
- eliminację błędów przy wykonywaniu dokumentacji zamówień.

Następne etapy procesu wytwórczego, przewidziane do automatyzacji, to:

- potwierdzanie zamówień
- wykonywanie kalkulacji
- wypisywanie rachunków i ofert
- sterowanie terminami
- sterowanie wykonawstwem (obrabiarki sterowane numerycznie).

* * *

Tablice decyzyjne stanowią dzisiaj na świecie nie tylko przedmiot zainteresowań teoretyków. Różnorodne problemy inżynierskie można z ich pomocą przedstawiać w sposób łatwo zrozumiały również dla osób bez informatycznego przeszkolenia.

Problem sformułowany za pomocą tablic decyzyjnych jest jednocześnie bardziej precyzyjny, co znakomicie ułatwia komunikację projektanta — inżyniera z systemem cyfrowym, którym musi się on coraz częściej posługiwać, jako podstawowym czynnikiem wzrostu wydajności procesów projektowania i wytwarzania.

WYKAZ LITERATURY

- [1] EGLER J. F.: Procedure for Converting Logic Table Conditions into an Efficient Sequence of Test Instruction. Communications of the ACM, 1963 vol 6, nr 9.
- [2] KIRK H. W.: Use of Decision Tables in Computer Programming. Communications of the ACM, 1965, vol 8, nr 1.
- [3] PRESS L. J. Conversion of Decision Tables to Computer Programs. Communications of the ACM, 1965 vol 8, nr 6.
- [4] POLLACK S. L.: Conversion of Limited-Entry Decision Tables to Computer Programs. Communications of the ACM, 1965, vol 8, nr 11.

Perspektywy powszechnej komputeryzacji w USA

Przedstawiono amerykańskie programy rozwoju zastosowań informatyki w administracji państwowej, regionalnej i municypalnej. Omówiono plany stworzenia ogólnokrajowego systemu transmisji danych w USA oraz wyniki badań prognostycznych nad niektórymi skutkami społecznymi nowych systemów komunikacyjnych.

Informatyka w administracji

Rozwój zastosowań informatyki w USA odbywa się w sposób odmienny niż w Japonii, Francji, Belgii czy innych krajach, gdzie interwencjonizm rządu wynika z określonego rządowego programu rozwoju informatyki¹⁾. U podstaw polityki rozwoju informatyki w tym kraju leży: formułowanie norm unifikujących rozwiązania systemowe i sprzętowe, określenie zasad i popieranie zakupów sprzętu dla agencji rządów stanowych i federalnego, intensywne szkolenie użytkowników, finansowanie badań i prac rozwojowych, wielokierunkowe rozwijanie systemów informatycznych w administracji (finansowanych z budżetu na zasadach rachunku ekonomicznego), stosowanie ulg podatkowych wobec prywatnego sektora inwestującego w sprzęt informatyczny. W rezultacie pomimo braku generalnego programu budowy ogólnokrajowych systemów informatycznych, systemy takie powstają tam szybciej niż w innych krajach.

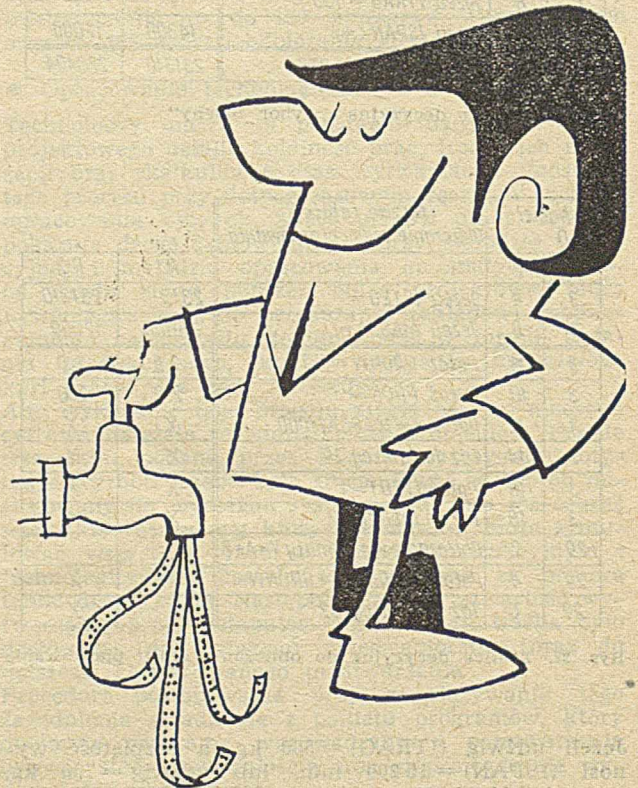
Pierwszy użytkowany komputer amerykański — ENIAC — został zbudowany w 1946 roku z funduszy rządowych i przekazany wojsku, natomiast pierwszy komputer do przetwarzania danych — UNIVAC I — został zakupiony w 1951 roku przez Urząd Statystyczny. Administracje federalna i stanowa dysponują 10% parku wszystkich komputerów zainstalowanych w kraju, co w roku bieżącym stanowi ponad 10 tys. maszyn o wartości ok. 3,5 mld dol. Wśród kilkudziesięciu dostawców sprzętu przeważają firmy IBM (26% dostaw) oraz UNIVAC (20%)²⁾.

W 1970 roku opracowano w USA 10-letni Program Rozwoju Informatyki w Administracji na lata 1970—1980³⁾. Jego realizację koordynują trzy agencje. Całość problemu planuje i koordynuje Biuro Zarzą-

dzania i Budżetu Urzędu Prezydenckiego. Biuro to prowadzi co roku aktualizację 5-letniego planu rozwoju informatyki oraz kontroluje jego wykonanie we wszystkich ogniach administracji. Zakupy sprzętu prowadzi Biuro Administracji Usług Urzędu Prezydenckiego (resorty są upoważnione do samodzielnego zakupywania sprzętu tylko do ceny 50 tys. dol., natomiast pozostałe zakupy muszą być zatwierdzone centralnie, z zachowaniem ustalonych zasad rachunku ekonomicznego). Krajowe Biuro Norm prowadzi prace normalizacyjne, badawcze i rozwojowe dla potrzeb administracji, głównie w zakresie rozwiązań wspólnych dla różnych agend administracji.

System transmisji danych

Sprawą pierwszej wagi, podnoszoną w wymienionym Programie jest problem transmisji danych. Oszacowano⁴⁾, że w USA udział systemów teleinformatycznych (*on-line*) w ogólnej liczbie systemów informatycznych wynosił w 1965 roku 1%, w 1970 roku —

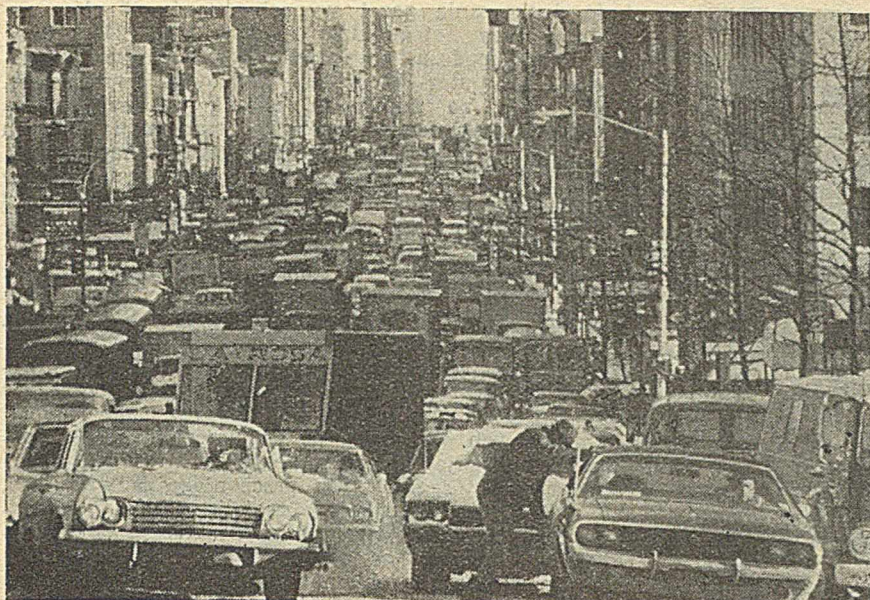


¹⁾ Główny motyw tych programów dotyczy ochrony krajowego przemysłu informatycznego przed dostawami sprzętu amerykańskiego.

²⁾ Według General Services Administration (1970) stan parku kształtował się następująco: 1950 — 2; 1952 — 5; 1954 — 10; 1956 — 90; 1960 — 531; 1962 — 1030; 1964 — 1862; 1966 — 3007; 1968 — 4232; 1970 — 5300.

³⁾ B. Gilchrist, M. Wessel — Government regulation of the computer industry, IFIPS Press 1972, s. 27.

⁴⁾ Walker F. Bauer, „On-line systems — The Characteristics and Motivations”. Proceedings at UCLA, Informatics Symposium on On-line Computing Systems (1965).



50%, a w 1975 r. ma osiągnąć 90%. W 1966 roku problematyką transmisji danych zajęła się Federalna Komisja d/s Łączności, która w marcu 1971 roku wydała dokument dotyczący zasad współpracy między przemysłem informatycznym a przedsiębiorstwami łączności. W dokumencie tym między innymi stwierdzono, że⁹⁾:

- a) sprzedaż usług informatycznych (z zakresu przetwarzania danych) nie powinna wpływać na ekonomiczną efektywność usług łączności
- b) koszty związane ze świadczeniem usług nie powinny ani bezpośrednio ani pośrednio obciążać użytkowników łączności
- c) dochody z usług łączności nie powinny być wykorzystywane na finansowanie usług informatycznych
- d) usługi informatyczne i usługi łączności muszą podlegać zasadom wolnej konkurencji między przemysłem informatycznym a przedsiębiorstwami łączności.

Ponadto wspomniana Komisja ustaliła podział usług łączności na rzecz systemów informatycznych w następujący sposób:

- przesyłanie komunikatów po cenie określonej w zależności od rozmiarów przesłanej informacji; cena ustalona jest przez przedsiębiorstwo łączności, w którego interesie leży stosowanie takich rozwiązań technicznych, które czyniłyby usługi tego typu opłacalnymi.
- dzierżawienie użytkownikom łączy telekomunikacyjnych; użytkownik zainteresowany jest w takim zorganizowaniu transmisji, aby w dzierżawionym odciunku czasu mógł przesłać jak największą liczbę informacji.

Powyższe zasady są dość charakterystyczne dla tego kraju, w którym ogólnokrajowy system informatyczny zaczął rozwijać się przede wszystkim w zakresie ogólnokrajowego systemu transmisji danych. System ten został wyposażony w ogólnodostępne urządzenia masowej pamięci i sprzęt obliczeniowy. Koncern WESTERN UNION pierwszy zorganizował tego typu system, kiedy to w 1966 roku otworzył w Nowym Jorku centralny ośrodek, będący częścią tzw. „Krajowej informacji użyteczności publicznej” (*national information utility*).

⁹⁾ B. Gilchrist, M. Wessel: Government regulation of the computer industry, AFIPS Press, 1972.

W 1973 roku Federalna Komisja Łączności podjęła historyczną decyzję w sprawie utworzenia ogólnokrajowego systemu transmisji danych¹⁰⁾. Zorganizowanie systemu zostało powierzone firmie PACKET COMMUNICATIONS Inc. (PC), którą założyli byli współprojektanci sieci ARPA. System ma polegać na zainstalowaniu do 1978 roku w 57 miastach USA (w 26 miastach do 1975 — procesorów pośredniczących w wymianie informacji (*Interface Message Processor*) — IMP pomiędzy różnymi komputerami oraz urządzeniami końcowymi innych sieci transmisji danych. System zbudowany jest w oparciu o łącza telekomunikacyjne typu ATT, przesyłające 1000-bitowe porcje danych z prędkością 50 Kb/s.

Do największych amerykańskich sieci teleobliczeniowych należą:

- MARK III (General Electric), która świadczy usługi w 250 miastach świata; łączna wartość jej usług wyniosła do 1973 roku 1,5 miliona „użytkownikogodzin”.
- CYBERNET (CDC) o łącznej długości łączy 38 tys. km, z których 6,4 tys. km charakteryzuje się szybkością 50 Kb/s. Obecnie sieć ta jest przekształcana w kierunku wyposażenia jej w 3 wielkie procesory, między którymi przesyłanie danych odbywać się będzie z prędkością 250 Kb/s. Wokół każdego tego typu procesora będą utworzone podsieci z obecnej sieci CYBERNET. Dla nowej sieci przyjęto nazwę COM-SOURCE
- ARPANET łącząca 40 komputerów, na których realizowane są systemy pilotowe zlecone przez administrację
- OTCOPUS (The Lawrence Livermore Laboratory), w której skład wchodzi jeden komputer CDC 6600, trzy CDC 7600, jeden XDS Sigma 7, dwa PDP-10 i CDC Star-100. Wszystkie wymienione komputery (prócz ILLIACA IV) są najszybszymi, jakie kiedykolwiek wyprodukowano. Zainstalowane są one w jednym budynku, tworząc ośrodek o największej mocy obliczeniowej na świecie
- AUTODIN (Departament Obrony), sieć której dzienna przepustowość wynosi 600 mln słów. Wymienione i inne sieci obliczeniowe, dzięki decyzji Federalnej Komisji Łączności, będą mogły ze sobą współpracować.

Krajowy system transmisji danych zapoczątkuje lawinę innowacji w zakresie systemów opierających się na łączności. Jeśli podzielić dochody wszystkich rodzajów publikatorów¹¹⁾ przez liczbę domostw, wy-

	rocznie	miesięcznie
	dol.	dol.
telefon	225	19
gazety	120	10
poczta	116	10
TV	102	8
periodyki	44	4
książki	42	3
radio	26	2
płyty	13	1
razem	688	57

¹⁰⁾ R. Malik: 26-city link system gets the go-ahead. Computer weekly, 10.I.1974 r. s. 8.

¹¹⁾ tzn. środków masowego przekazu oraz telefonu i wydawnictw książkowych.



datki tych ostatnich w 1969 roku były następujące⁸⁾: Przewiduje się, że wydatki te wzrosną do roku 1985 o 40%, a w roku 2000 wyniosą minimum 1400 dol. rocznie (przy wzroście przeciętnych dochodów do 22 000 dol.). Obecna forma usług pocztowe wymagające różnego typu druków (papieru) będą zastąpione przesyłaniem sygnałów elektronicznych między nadawcą a odbiorcą. Posługiwanie się gazetami może ulec zmianie, polegającej na wywoływaniu na ekranie domowego telewizora potrzebnej informacji. Prę-

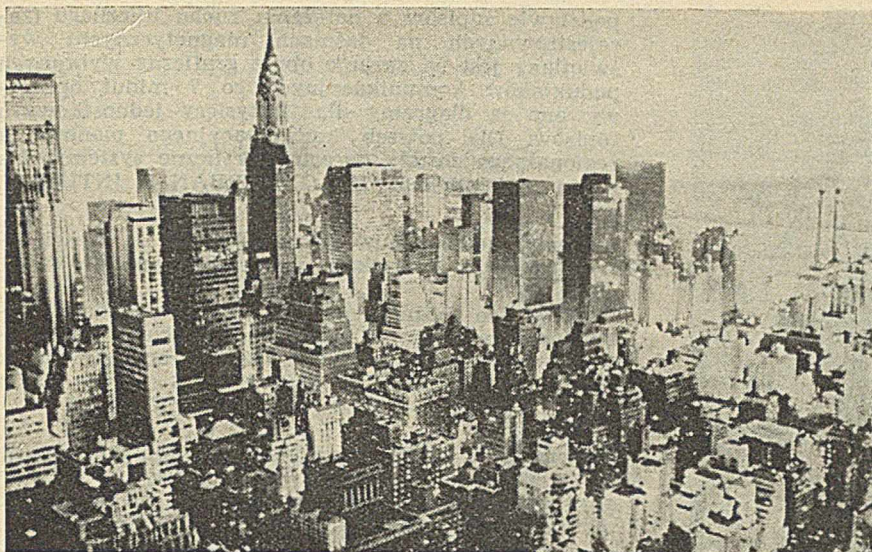
kość drukowania gazet wynosi obecnie 45 słów na minutę, przy fotoskładzie dochodzi do 2400 słów na minutę, podczas gdy prędkość w przesyłaniu końcówka—komputer—komputer—kończówka wyniesie ok. 86 000 słów na minutę. Z chwilą upowszechnienia telewizji kablowej, szybkość transmisji danych zwiększy się 1000-krotnie w stosunku do łączy przesyłających głos. Wówczas komputery będą zdolne gromadzić i wyszukiwać informacje z prędkością 12 milionów słów na minutę, przekształcając je w użytkową formę na mikrofilmie z prędkością 0,7 mln słów na min. skąd druk dokumentu będzie odbywał się w tempie 0,18 mln słów na min. Ten ocean słów nie pozostanie bez wpływu na osobowość człowieka, który jest w stanie przetwarzać 250 słów na min. (człowiek o wysokich kwalifikacjach używając metody tzw. szybkiego czytania, może przetwarzać 1000 słów na min.). Amerykańskie Stowarzyszenie Agencji Reklamowych wyliczyło, że do przeciętnego Amerykanina adresuje się dziennie 1600 informacji handlowych, z których 80 jest przez niego zauważonych, ale tylko 12 powoduje jego reakcję. Nawał informacji niesie ze sobą ogromne niebezpieczeństwo znieczulenia człowieka na bodźce płynące z jego otoczenia. Rozwiązaniu problemów technicznych przepływu informacji towarzyszyć więc muszą stałe badania psychospołeczne nad możliwościami dopasowania systemu nerwowego człowieka do nowoczesnego systemu komunikowania.

Ogólnokrajowy system transmisji danych umożliwi szybsze, niż się spodziewano, uruchomienie systemu końcówek domowych. Szczegółowe badania w tym zakresie zostały wykonane w 1972 roku przez firmę BELL (Kanada)⁹⁾. Badania te oparto na koncepcji substytucji przez końcówki domowe obecnego systemu komunikowania się wewnątrz miasta i pomiędzy miastami w zakresie prac biurowych i domowych. Z badań tych wynika, że biura będą stopniowo w latach 1976—1980 wyposażone w audio-wizualne i informatyczne środki prowadzenia na odległość konferencji i rozmów. Zdalne ośrodki pracy w domu będą prawdopodobnie eksperymentowane w latach 1981—1990. Sąsiedzkie (osiedlowe) środki zdalnej pracy będą powstawać około 1986—1990 roku. Mobilne końcówki do specjalnych zadań będą stosowane około 1986—1990 roku. Szansa stosowania tego typu rozwiązań prawdopodobnie nastąpi przed końcem obecnego stulecia, przy czym główną przeszkodę w ich rozpowszechnianiu będą stanowiły aspekty społeczne, a nie technologiczne.

⁸⁾ Bagdikian: Mass Communications Computers and the problems of society AFIPS PRESS 1972 Montval, New Jersey s. 231.

⁹⁾ L. H. Day: The future of computer and communications services. IFIP 1973.





Badania nad systemem komunikowania się między miastami wskazują, że w kierowaniu najbardziej pożądanym jest osobisty kontakt oraz że większość niezbędnych tu rozwiązań technicznych jest już obecnie dostępnych. Należą do nich: dwukanałowa (ze sprzężeniem zwrotnym) telewizja, telekomputerowy system prowadzenia konferencji, a dalsze ulepszenia będą prowadziły do stosowania ściennych (płaskich) ekranów telewizji kolorowej.

W zakresie prac domowych zostały przeprowadzone badania nad możliwością wykorzystania końcówek przez ekspertów — specjalistów i ekspertów — typu gospodynie domowe. Zostały wyróżnione: usługi w pełni konwersacyjne z komputerem i usługi o ograniczonych możliwościach konwersacji. Do pierwszej grupy zaliczono: zdalne zakupy, zdalne rachunki bankowe, elektroniczny system zabezpieczenia domu, nauczanie programowane. Do drugiej grupy zaliczono: porady w zakresie zakupów i usług, serwis informacji prasowej, serwis wolnych miejsc pracy, serwis usług szkoleniowych, serwis poszukiwania osób, domowe usługi obliczeniowe. Odnośnie pierwszego rodzaju usług, opinia dotycząca celowości ich zastosowania sprowadza się do tego, że o ile będą one nieodpłatne lub w granicach „mniej niż 1 dol./m-c” to można je zaakceptować. Stwierdzono, że obecne rozwiązania w tym zakresie są wystarczające. Drugi typ usług wnosi istotne zmiany w ich obecnym sposobie funkcjonowania, lecz pomimo tego pozytywne opinie otrzymał tylko serwis wolnych miejsc pracy i dostępnych form szkolenia.

Oceniono, że proponowane rozwiązania w „Mieście drutów” uczynią z miasta elektroniczne więzienie ze zdepersonalizowanym rynkiem. Szereg wymienionych czynności domowych ma charakter rekreacyjny i — jak jeden z respondentów stwierdził — że „czasami jest konieczne, aby nie były one zbyt dobrze zorganizowane”. W konkluzji badań stwierdzono, że nowe możliwości systemu komunikowania będą prowadziły raczej w kierunku ewolucji dotychczasowych systemów, a nie w kierunku rewolucji elektronicznej.

Centralne systemy informatyczne

Koncepcja centralnego państwowego systemu informatycznego sprowadza się do wykorzystywania informatyki w zakresie przygotowywania rządowych posunięć dotyczących szczególnie wzrostu gospodarczego, zatrudnienia, ekonomiki zdrowia, szkolnictwa, opieki społecznej i zasiłków. Zastosowano modele ekonometryczne w celu prognozowania i wyjaśniania związków między rozwiązaniami monetarno-fiskalnymi, a ich strukturalnym wpływem na kształtowanie się dochodu narodowego, cen, bezrobocia itp. Główny-

mi użytkownikami są takie instytucje, jak Krajowe Biuro Badań Ekonomicznych, Rada Doradców Ekonomicznych, Departament Handlu, Biuro Statystyki Pracy, Biuro Rezerw Federalnych, Biuro Zarządzania i Budżetu i inne jednostki. Rozwój metod informatycznych rządu amerykańskiego idzie w kierunku weryfikowania hipotez, a nie ich formułowania, co spowodowało rozwój metod symulacyjnych. Ten zakres metod jest wykorzystywany głównie w zakresie badania zależności wysokości podatków i dochodu narodowego oraz w zakresie analizy środków stabilizujących gospodarkę¹⁰⁾.

Regionalne systemy informatyczne

Szczególne miejsce w USA zajmuje problematyka systemów informatycznych dla potrzeb miast. Samodzielnych jednostek miejskich jest w USA ok. 20 tys. a budżet ich sięga setek miliardów dolarów. W ramach tych budżetów, w granicach od 0,5 do 1,5% finansowane są wydatki na rozwój i utrzymanie systemów informacyjnych i informatycznych. Systemy informatyczne projektowane są dla potrzeb:

- usług miejskich, technicznych, administracyjnych i dla ludności. W zakresie usług technicznych stosowane są systemy sterowania procesami: dostaw wody, gazu, elektryczności, transportowymi (obserwacja dróg, autostrad, sterowanie ruchem ulicznym, komunikacją miejską), ochrony środowiska, budowy mostów i obiektów, melioracji, kontroli sytuacji powodziowych.

- usług administracyjnych: kontroli podatków, gospodarki terenami, planowania i kontroli budżetu, prowadzenia rejestrów urodzin i zgonów, list uprawnionych do głosowania, gospodarki mieszkaniowej, planowania rozwoju ekonomicznego regionu, zaopatrzenia administracji, ochrony konsumenta.

- usług dla ludności: szkolnictwa, pogotowia lekarskiego, przeciwpożarowego, policyjnego, wieziennictwa, nadzoru budowlanego i sanitarnego, utrzymania parkingów, ulic i dróg rekreacyjnych, opieki nad dzieckiem, opieki społecznej i sądownictwa, obsługi komunikacji miejskiej, opieki lekarskiej (kliniki, szpitale, ośrodki zdrowia, usługi pielęgniarskie, kontrola leków itp.) Wymienione systemy informatyczne są projektowane w czterech wersjach: systemy informacyjne (komputerowe), systemy grafiki komputerowej, systemy urbanistyczne, systemy gier.

Do najbardziej zaawansowanych systemów informacyjnych (komputerowych) zalicza się systemy stosowane przez władze miejskie w Monroe — Luisiana (gdzie osiągnięto znaczne efekty ekonomiczne), w Nassau County — New York (gdzie uruchomiono międzywydziałowy system raportów), w Charlotte N.C. i Wichita Fall — Texas.

Z punktu widzenia potrzeb municypalnych systemów informacyjnych został zaprojektowany w 1970 roku system spisu powszechnego. Do ciekawszych podsystemów można zaliczyć ADMACH (Opracowany w ramach Południowo-Kalifornijskiego Regionalnego Systemu Informacyjnego — SCRIS), który umożliwia łatwe łączenie informacji dotyczących jednego adresu z różnych kartotek.

Podobny podsystem pod nazwą DIME (*Dual Independent Map Ecoding File*), opracowany przez Biuro Statystyczne, umożliwia cyfrowe kodowanie terenów (ulic itp.) i jest częścią składową systemu CRAM (*Computerized Resource Allocation Model*), który wykorzystywany jest dla potrzeb ustalania lokalizacji nowych obiektów. Podsystem DIME zakłada i utrzy-

¹⁰⁾ Ch. Wolf, J. Enns: *Computers and Economics: Progress, Problems and Prospects. Computers and the problems of society. AFLEPS 1972, s. 143.*



muje bazę danych, a podsystem UNIMATCH organizuje różnego rodzaju operacje wyszukiwania i łączenia (ADMATCH).

Połączenie podsystemu informacji statystycznej z innymi rodzajami informacji zostało przykładowo rozwiązane w Biurze Analiz Miejskich w Los Angeles i w ramach projektu CLETR, w ośrodku regionalnym Cincinnati i Hamilton, Ohio. Projekt CLEAR ma podsystemy przystosowane do obsługi systemu informacji federalnej, stanowej, miejskiej i powiatowej. Dla potrzeb planowania regionalnego został zaprojektowany system URBANDOC (City University, New York), który zbiera informację tematyczną z dokumentów, książek i artykułów.

Systemy grafiki komputerowej są wykorzystywane przy opracowywaniu map, prezentowaniu danych demograficznych, statystycznych, itp. Ośrodek Obliczeniowy Uniwersytetu opracował system opracowywania map demograficznych (SYMAP).

System CARTOGRAPHATRON został wykorzystany w badaniach obciążenia komunikacji w Chicago. Na

podstawie zapisów o natężeniu ruchu ulicznego (zarejestrowanych na taśmach magnetycznych) wyświetlany jest na ekranie obraz graficzny wybranych podkładów komunikacyjnych co 7 minut opracowywane są diagramy dla 20 tysięcy jednostkowych zapisów. Dla potrzeb konwersacyjnego planowania regionalnego zostały opracowane liczne systemy, jak m. in. DISCOURSE, IMAGE, URBAN 5, INTU-VAL i inne.

Systemy modelowania urbanistycznego są rozwiązywane przede wszystkim metodami symulacyjnymi.

Przykładowo można wymienić symulację ruchu ambulansów pocztowych w Nowym Yorku. Do najbardziej znanych systemów należą opracowane przez firmę RAND-model METROPOLIS oraz przez J.W. Forrestera — model URBANDYNAMICS.

Systemy gier służą za narzędzie doskonalące działania planistów i decydentów. Do bardziej popularnych gier ilustrujących problematykę decyzji miejskich można zaliczyć: CITY I i II, METRO, APEX.

W celu skordynowania rozwoju municypalnych systemów informatycznych postuluje się utworzenie Instytutu Miasta na wzór Instytutu Erokharena i Salika. Warto wspomnieć, że już w 1963 roku zostały podjęte przez firmę RAND badania nad unifikacją municypalnych systemów informatycznych. Wprowadzono nawet nazwę „zunifikowany system informacyjny”¹¹⁾.

Formułowane są opinie, że rozwój omawianych systemów informatycznych wymaga ujednoczenia form przesyłania informacji między jednostkami municypalnymi i federalnymi¹²⁾. Pod tym względem najlepiej rozwijane są systemy informatyczne dla potrzeb policji. Federalne Biuro Śledcze (FBI) zorganizowało Krajowy Ośrodek Informacji Kryminalnej (NCIC), który udostępnia informacje na temat poszukiwanych osób i kradzieży samochodów. Trzy tysiące zainstalowanych stanowych i miejskich komendach policji końcówek każdego dnia przetwarza ponad 4 tys. komunikatów. Podejmowane są prace nad integracją systemów w zakresie banku danych o dzieciach robotników (zmieniających stale miejsce zamieszkania) oraz w zakresie krajowej sieci informacyjnej dotyczącej systemu opieki nad rodziną. Wśród wysuwanych argumentów na rzecz integrowania systemów informatycznych, wybijają się stwierdzenie, że posługiwanie się informacjami dotyczącymi tej samej tematyki, ale przygotowywanymi w ramach różnych, nieujednoczonych systemów informatycznych, może prowadzić do błędnego wnioskowania i rozwijania się „datamani”.

Jednym z rozwiązań o charakterze integracyjnym miało być Krajowe Centrum Danych o Obywatelu. Projekt utworzenia tego centrum wywołał olbrzymią dyskusję nie tylko w Kongresie, ale i wśród prawników w innych środowiskach zawodowych. Projekt uznano bowiem za naruszający wolność człowieka. W rezultacie, w jego miejsce, ma być utworzony Federalny Ośrodek Analiz Statystycznych, którego spokojniejsza nazwa umożliwi spełnienie niektórych zadań stawianych przed Krajowym Centrum Danych.

¹¹⁾ E. Hearle, R. Mason: A data processing system for state and local government. Prentice-Hall, Inc. Englewood Cliffs.

¹²⁾ A. Miller: The Assault on Privacy. Computers data banks and dossiers. The University of Michigan Press, 1971.

dok. ze str. 11 wykazu LITERATURY do art. B. Burgielskiego

[5] VEINOTT C.: Programming Decision Tables in FORTRAN, COBOL or ALGOL. Communications of the ACM, 1966, vol 9, nr 1.

[6] CHARMAN A. E. M. D.: Description of Basic Algorithm in Detab/65 Preprocessor. Communications of the ACM, 1967, vol 10, nr 7.

[7] PERTILLER H., MATHIES H., VEISMANN A.: Technische Angebote und Auftragsbearbeitung in der Varianten — Fertigung mit Entscheidungstabellen.

[8] ERBACH K. F.: Entscheidungstabellen — Theorie und Praxis. Computer — Praxis, 1972 nr 12.

[9] NIEDZWIĘCKI J.: Tablice decyzyjne — struktura i zastosowanie w: INFORMATYKA, 1972, nr 1.

Dr JAN M. SZYMAŃSKI od chwili ukończenia studiów prawniczych (1951 — dyplom zawodowy, 1958 — magisterski) łączył pracę zawodową radcy prawnego z pracą naukową, zajmując się m.in. problematyką uściślenia kodowania norm prawnych i usprawnienia procesów sterowania społeczeństwem przy pomocy prawa. Uzupełniając studia matematyczne indywidualnie i na kursach zastosowań matematyki PAN, uzyskał doktorat nauk prawnych na Uniwersytecie Śląskim (1970). Jest Przewodniczącym Sekcji Nauk Społecznych łódzkiego oddziału Polskiego Towarzystwa Cybernetycznego i wykładowcą na kursach informatyki. Pracuje nad podstawami teoretycznymi informacji, decyzji i sterowania w odniesieniu do układów społecznych. Aktualnie jest zatrudniony w Łódzkim Kombinacie Budowy Domów, jako radca prawny i konsultant w zakresie informatyki.



JAN M. SZYMAŃSKI

Łódzki Kombinat Budowy Domów

681.3.(438)

W sprawie elementarnego wykładu informatyki

Autor zwraca uwagę na pewne wady występujące w szeroko prowadzonej w Polsce działalności dydaktycznej w dziedzinie informatyki. Przedstawia uproszczoną koncepcję struktury elementarnego wykładu podstaw informatyki.

Postępy nauk matematyczno-przyrodniczych w pierwszej połowie XX wieku sprawiły, że wielu specjalistów z zakresu nauk społecznych począł nurtować problem stworzenia uniwersalnej, interdyscyplinarnej wiedzy o społeczeństwie, jako o zorganizowanym systemie przekształceń¹⁾.

Do dalszej racjonalizacji nauki o społeczeństwie zdaje się również nawoływać dramatyczny apel zawarty w końcowych zdaniach obiegającego cały świat raportu rzymskiego²⁾.

Zagadnienie zintegrowania i optymalnego dostosowania do potrzeb nauk społecznych takich dyscyplin i metod, jak badania operacyjne, statystyka matematyczna i symulacja, teoria procesów stochastycznych, ogólna teoria systemów, metody analizy systemowej itp., stanowi zadanie dla całego pokolenia teoretyków, zajmujących się matematyzacją nauk społecznych. Istnieje jednak dziedzina nieodwracalnego postępu, w której można by już dziś pozwolić sobie na metodologiczną racjonalizację. Dziedziną tą jest informatyka.

Obecnie bez mała cała Polska studiuje informatykę. Za pomocą telewizji i bez niej, ze skryptów i notatek, na kursach stacjonarnych i zaocznych, zbiorowo i indywidualnie. Na podstawie doświadczeń własnych

i cudzych w sferze nauczania informatyki doszedłem do pewnych refleksji, którymi chciałbym się podzielić w tym artykule.

Nie jest moim zamiarem upraszczanie złożonej problematyki programów nauczania. Celowo ograniczam się do zwrócenia uwagi na pewne wady i braki wykładu informatyki, dość rozpowszechnione i niezależne od treści i zakresu programów.

Zdarza się, że wykład informatyki (nawet uniwersytecki) sprowadza się do podania konglomeratu odczuwanych pojęć i charakterystyk technicznych, przydatnych jedynie dla praktyka, mającego już opanowane podstawy przedmiotu, ale zupełnie nie poddających się percepcji słuchacza surowego, zwłaszcza, jeżeli słuchacz ten dysponuje tylko wykształceniem średnim lub humanistycznym. Przystawalność takiego wykładu osłabia jeszcze dodatkowa okoliczność, że z reguły słuchaczom nie wyjaśnia się, lub wyjaśnia się w sposób nie dostateczny terminy matematyczne lub techniczne, nie zawsze zresztą jednolicie tłumaczone z języka angielskiego. Brak jednoznacznych norm terminologicznych jest nieuniknionym zjawiskiem towarzyszącym każdej dyscyplinie, której nie zdążyła jeszcze ukształtować tradycja naukowa. Nie zwalnia nas to jednak z obowiązku dokładania wszelkich starań o przybliżenie podstawowych pojęć do możliwości percepcyjnych przeciętnego słuchacza.

Rozmawiałem z inżynierami i ekonomistami, słuchaczami wykładów informatyki, którzy ukończyli studia przed 9—15 laty. Wszyscy uskarżali się na niedostateczną zrozumiałość takich zagadnień, jak: pojęcie miary entropii źródła i jego praktyczna użyteczność, operowanie układami równań liniowych i macierzami algebraicznymi, optymalizacja drogą programowania liniowego itp. Okazuje się, że ta część wiedzy wchłoniętej w okresie studiów, która nie znalazła zastosowania w późniejszej praktyce zawodowej, zatarła się w ich pamięci prawie całkowicie. Odświeżyć tę pamięć nie jest naogół rzeczą trudną, pod warunkiem, że wykładowca tego chce, a jed-

1) por. M.I. Sietrow: Podstawy metodologiczne budowy jednolitej teorii organizacji, PRAKSEOLOGIA 1970 nr 36 s. 67 i 69.

2) D.H. Meadow i in.: Granice wzrostu (The Limits to Growth) PWE 1973 r. s. 206.

nocześnie potrafi wyzwolić słuchacza z klimatu oniesmielenia.

Ponadto autorzy skryptów czy wykładowcy nie odróżniają z reguły tego, co jest już jednoznacznie przyjęte w nauce lub praktyce — od swoich własnych i cudzych, odosobnionych nieraz, poglądów. Utrudnia to słuchaczom dodatkowo przyswojenie sobie wymaganego minimum wiedzy i uzyskanie niezbędnego stopnia pewności.

Następnym uchybieniem wobec czytelnika lub słuchacza bywa niedostateczne wyjaśnianie pojęć podstawowych, których mętne zarysy utrudniają budowanie złożonych systemów pojęciowych i ich samodzielne przekształcanie. Tak np. bardzo rzadko się zdarza, aby wyjaśniono słuchaczom istotę i różnicę pojęcia informacji w znaczeniu semantyki języka naturalnego i w znaczeniu matematycznym, ilościowym. A przecież przekształcanie pierwszej w drugą i na odwrót stanowi istotę użyteczności maszyn cyfrowych. Trzeba się tu jednak pilnie wystrzegać zaplątania w dywagacje teoretyczne, które mogą spowodować np. nadużywanie pojęcia izomorfizmu, które zachowuje całkowity sens na gruncie matematyki, a poza jej granicami powinno być traktowane z dużą ostrożnością, zwłaszcza tam, gdzie nie mamy do czynienia w sposób oczywisty ze zmiennymi ilościowymi, lub gdzie agregacja informacji prowadzi do homomorfizmu.⁴⁾

Najpoważniejszym grzechem wobec słuchacza wydaje się być jednak brak racjonalnej struktury wykładu, tzn. takiej, która by pozwalała mu optymalnie budować gmach poznania. W świetle twierdzeń psychologii, podstawowym warunkiem efektywnego uczenia się jest stworzenie szkieletu pojęciowego, który w procesie dalszego uczenia się mógł by być rozbudowany i przekształcony. Na podstawie prze-myśleń własnych i przeprowadzonych sondaży, określiłbym następująco zarys struktury elementarnego wykładu z zakresu podstaw informatyki:

- pojęcia elementarne
- tradycyjne systemy informacyjne
- budowa i działanie komputera
- zautomatyzowane systemy informacyjne
- przygotowania do wdrażania systemów informatycznych.

Jest to koncepcja maksymalnie uproszczona, bez żadnych rewelacji. Ale zawiera ona w sobie jednak elementy zarówno nauk społecznych jak i technicznych, co wymaga od kadry wykładowców zdania sobie sprawy z faktu, że interdyscyplinarny charakter informatyki wymaga od każdego „mono-specjalisty” dodatkowego uzupełnienia wiedzy. Poniższe uwagi — omawiające pokrótce kolejne punkty proponowanej struktury wykładu — mają luźny charakter i bynajmniej nie pretendują do wyczerpującego omówienia problematyki.

Pojęcia podstawowe

Przyswojenie sobie przez słuchaczy podstawowych pojęć informatyki jest warunkiem sine qua non dalszego poznania dziedziny. Nie powinno się to jednak sprowadzać do elektrycznego przeglądu oderwanych pojęć, na podobieństwo kartkowania „słownika cybernetycznego”. Te podstawowe pojęcia stanowią powinny fundament wiedzy, spojony więzią systemową.

3) E. Kofler: O wartości informacji, PWN 1968; M. Mazur: Jakościowa teoria informacji, WT 1970.

4) por. „Modelowanie jest w świetle naszych rozważań złożonym z działań procesem konstruowania modelu izomorficznego względem oryginału, to jest takiego modelu, o którym można powiedzieć, że... a) jest uproszczonym odbiciem oryginału (a więc nie jest wierną jego kopią...)”. J. Gościński: Cybernetyczne podstawy informatyki, str. 29, Telewizyjny Kurs Informatyki, wyk. B/2,3 OBRI.

Trzeba by zacząć od pojęcia informatyki, które zwykle zbywa się jednym zdaniem. Nie należy ukrywać metodologicznej nieściśłości tego pojęcia⁵⁾, wyjaśniając jego podwójne znaczenie: jako teorii zajmującej się zbieraniem, przetwarzaniem, przechowywaniem i rozpowszechnianiem informacji za pomocą ETO, oraz jako praktyki organizacyjnej gospodarowania informacją za pomocą współczesnych środków technicznych. Informatyka racjonalizuje łączność wśród systemów, których pojęcie również zasługuje na większą uwagę, niż się to na ogół czyni.

Przed wszystkim trzeba wyraźnie powiedzieć, że „system” jest pojęciem, które dziś zastępuje w znacznej mierze tradycyjne kategorie ontologiczne „osób”, „rzeczy” itp. (pisał o tym A. Targowski⁶⁾), ponieważ umożliwia nam interdyscyplinarne, ujednoczone podejście do zjawiska przekazywania informacji w przyrodzie i społeczeństwie — czego wymaga od nas informatyka. Uświadomiwszy to sobie, słuchacze nie będą zgadywać, który z elementów otaczającej ich rzeczywistości gospodarczej jest „systemem”, a który nim nie jest. Trzeba zachować wyraźną granicę między pojęciem, a desygnatem.

Następnie trzeba pamiętać, że w nauce współczesnej znanych jest więcej znaczeń określenia „system”. Najbardziej znane jest pojęcie systemu aksjomatycznego ogólnej teorii systemów L. von Bertalanffy'ego⁷⁾ oraz pojęcie cybernetyczne „czarnej skrzynki” Wienera, stosunkowo jasno zdefiniowane u nas w znanej książce H. Greniewskiego.

Nawet wielkie umysły nie dokonują czasem tego rozróżnienia⁸⁾, chociaż sam Bertalanffy wskazuje na zasadniczą różnicę „organizmistycznego” pojęcia ogólnej teorii systemów i „mechanistycznego” pojęcia systemu cybernetycznego⁹⁾. Otóż w informatyce całkowicie wystarcza proste i jasne, cybernetyczne pojęcie systemu.

Zjawiska informacji, kodowania i transformacji komunikatów w systemach społecznych stosunkowo najprzystępniej można przedstawić z semiologicznego punktu widzenia, ukazując, jak łączność między ludźmi przy postępującej technice przekazu informacyjnego prowadzi od najprostszych do najbardziej złożonych struktur społecznych.

Tradycyjne systemy informacyjne

Nie ma efektywnej działalności bez silnej motywacji. Aby zmobilizować niezbędną energię ludzką do budowy systemów informatycznych, trzeba najpierw należyście uzasadnić tę potrzebę. Nie jest to trudne, ponieważ tradycyjne systemy informacyjne mają wiele powszechnie znanych wad.

Dlatego też dobrze jest nazwać te wady po imieniu, odwołując się do osobistych doświadczeń słuchaczy:

5) „Jak każda dziedzina in statu nascendi, pojęcie informatyki nie zostało dotychczas ściśle zdefiniowane, a jego zakres pozostaje nadal dyskusyjny. Z tego względu wszelka próba dokładniejszego określenia przedmiotu i zakresu informatyki może być zakwestionowana w zależności od punktu widzenia, z którego rozpatrujemy tę dziedzinę. „Rocznik polityczny i gospodarczy”. PWN, str. 363.

6) A. Targowski: Informatyka, klucz do dobrobytu, PIW 1971 s. 53.

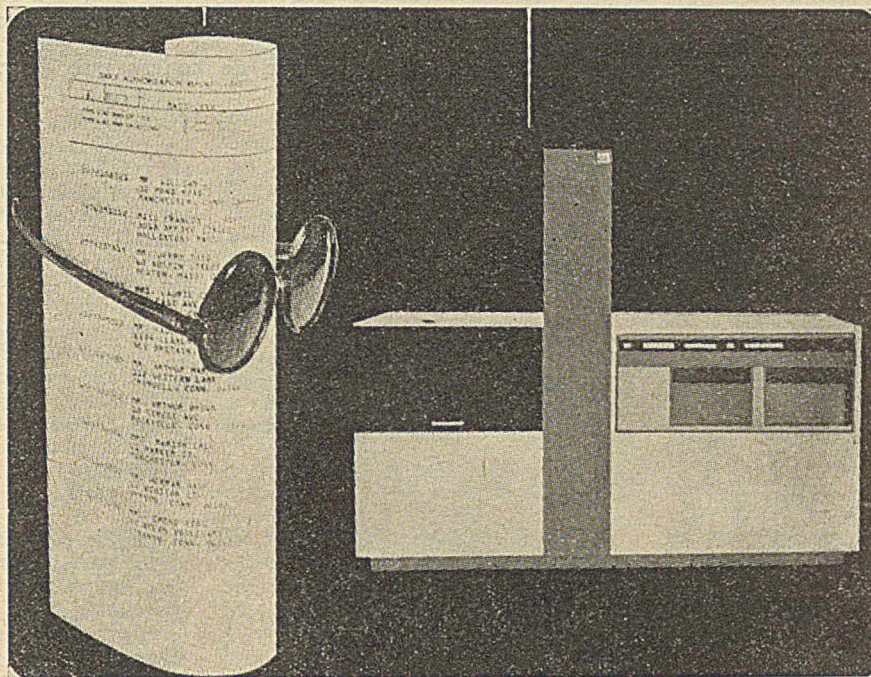
7) L. von Bertalanffy: The theory of open systems in physics and biology, Science 1950 nr III s. 23—29; An Outline of General System Theory, British Journal for Philosophy of Sciences 1950 z. I, s. 139—164 i in. Ostatnia najbardziej sytetyczna praca: General System Theory, NY Braziller 1968 tłum. niem. i włoskie.

8) Tak np. O. Lange, w pracy pt: Całość i rozwój w świetle cybernetyki PWN 1962, przyjął pojęcie systemu Bertalanffy'ego.

9) L. von Bertalanffy: Robots, Men and Minds, tłum. niem. pt: Aber von Menschen wissen wir nichts, rozdział „Die Kybernetik und ihre Grenzen” str. 111—122, a także 123, 124—126; rozdział „Offene system: die Flamme des Lebens” s. 128—139.

● **nadmiar informacji.** Każdy pracownik gospodarki uspołecznionej, zwłaszcza na stanowisku kierowniczym, odczuwa osobiście rosnący permanentnie nacisk informacji (problemów, zadań, decyzji). Wystarczy to tylko przypomnieć i uogólnić, jako rosnącą dysproporcję między napływającą, a skończoną i ograniczoną masą informacji, które statystyczny umysł ludzki zdolny jest efektywnie przetworzyć i zapamiętać.¹⁰⁾ Na ten nadmiar informacji składa się nadmiar w znaczeniu ścisłym, tzn. redundancja, oraz rosnąca ilość informacji absolutnie zbędnych.

● **niedobór informacji** występuje często w praktyce gospodarczej, będąc drugim, istotnym źródłem zakłóceń procesów zarządzania. Niedobór informacji może być często wynikiem ich zniekształcenia przez „szumy”.



● **informacja spóźniona.** Opóźniona informacja staje się entropią, należałoby więc wykazać, że maszyny zdolne są przyspieszyć proces dopływu i przetwarzania informacji w nieporównywalnym dotychczas stopniu¹¹⁾.

● **wadliwe przetwarzanie informacji,** tzn. błędy rodzące zniekształcenie informacji.

To zniekształcenie informacji może mieć różnorodne źródła np:

— wadliwy sematycznie lub syntaktycznie kod,
— wadliwość percepcji ludzkiej, spowodowana zarówno czynnikami fizjologicznymi (zmęczenie, zdernerwowanie), jak i psychologicznymi (amplifikacja i interpretacja przyjętej informacji stosownie do własnego systemu poglądów).¹²⁾

— świadome lub nieświadome pomijanie lub fałszowanie informacji szczegółowych, będących podstawą późniejszych decyzji.

Budowa i działanie komputera

Na wstępie należało by wyjaśnić zasady funkcjonowania pamięci maszyny, która przez tożsamość nazwy kojarzy się słuchaczom na ogół niepotrzebnie z pamięcią pojmowaną psychologicznie. Jako pozytywny przykład krótkiego i zwięzłego wyjaśnienia

problemu można wskazać popularną książeczkę prof. Pawlaka¹³⁾.

Następnie należałoby rozbudować wiedzę o pamięci maszyny, omawiając jej rodzaje i wzajemne zależności funkcjonalne oraz oprogramowanie współczesnych komputerów. W ten sposób słuchacz zostanie należycie przygotowany do zapoznania się ze szczegółową charakterystyką strukturalną i funkcjonalną kolejnych generacji technologicznych komputerów, przy czym najwięcej uwagi należało by poświęcić zastosowaniu maszyn do przetwarzania i wyszukiwania informacji z uwagi na najpowszechniejszy charakter tego rodzaju zastosowań.

Na zakończenie można sobie pozwolić na uwagi krytyczne, uświadamiając słuchaczom fakt, że maszyna nie jest lekarstwem na brak inteligencji lub inicja-

tyw, że nie może ona działać skutecznie w oparciu o wadliwe struktury organizacyjne i błędnie formułowane problemy¹⁴⁾. Można też naszkicować słuchaczom humanistyczną koncepcję zastosowań komputerów, jako dźwigni wyzwolenia ludzkości z jarzma własnych ograniczeń społecznych, w przeciwstawieniu do możliwości ich wykorzystania do skuteczniejszego ograniczenia i ujarzemia człowieka¹⁵⁾.

Zautomatyzowane systemy informacyjne

Przed przystąpieniem do omawiania projektowania zautomatyzowanych systemów informacyjnych (systemów informatycznych), wskazane jest zapoznanie słuchaczy z zagadnieniami elementarnymi w nieco szerszym zakresie niż to określono na wstępie. Dopiero ten słuchacz, który należycie poznał zasady budowy schematów blokowych przetwarzania, może orientować się w możliwościach zastosowania maszyn do obliczeń, symulacji i przetwarzania danych, w technicznych sposobach transmisji danych itd., będzie przygotowany do łączenia tych cząstkowych wiadomości w zorganizowaną całość.

10) por. Bzálawa: Nastawienie, podstawa regulacji psychicznej, PWN 1970 s. 191; J. Koziński: SOS z morza słów, KULTURA 1972 nr 11.

11) A. Targowski: Informatyka., ks. cyt. str. 5.

12) M. Maruszewski, J. Reykowski, T. Tomaszewski: Psychologia jako nauka o człowieku, KiW 1967 s. 304–305.

13) Z. Pawlak: Maszyny matematyczne, PZWS 1971.

14) por. A. Targowski j. wyż. s. 24; M. Mazur, Od emocji do decyzji, KULTURA 1971 nr 41.

15) por. art. o teorii informacji w/g FIGARO, FORUM z 1970 nr 51–52 s. 23, J. Weizenbaum: By informatycy stali się skromniejsi w/g DIE ZEIT, FORUM z 1972 nr 20; podobnie Targowski j. wyż. s. 61–62.

Przygotowania do wdrażania systemów informatycznych

Chciałbym tu przeciwstawić się ewentualnym zarzutom, że etap praktycznego wdrażania systemów wychodzi już poza zakres wykładu elementarnego. Informatyka jest przede wszystkim dyscypliną praktyczną, dlatego też doprowadzenie słuchacza do etapu praktycznego zastosowania nabytej wiedzy — chociażby tylko w bardzo ogólnym zakresie — powinno stanowić ukoronowanie całego kursu. Wiedza ta powinna składać się z wielu elementów, poczynając od określenia zasad analizy istniejących systemów informacyjnych, przygotowania danych, stosowania metod optymalizacyjnych itp., a skończywszy na zasadach eksploatacji systemu po odbiorze. Jak uczy praktyka, wprowadzanie maszyn cyfrowych wymaga zawsze w mniejszym lub większym stopniu przebudowy struktur organizacyjnych, co zawsze pociąga za sobą opór ludzi zainteresowanych w utrzymaniu istniejącego *status quo*. Opór taki tym

bardziej daje się we znaki, im większą grupę ludzi trzeba „ruszyć” oraz im wyżej uplasowali się oni w hierarchii organizacyjnej. Opór ten będzie tym mniejszy i łatwiejszy do pokonania, im większa liczba pracowników, zwłaszcza na stanowiskach kierowniczych, uświadamiać sobie będzie konieczności organizacyjne, kadrowe i psychologiczne, jakie niesie ze sobą wprowadzenie informatyki.

Konkludując stwierdzam, że na obecnym etapie rozwoju można by już zdobyć się na uporządkowanie podstawowej terminologii, używanej w dziedzinie informatyki, oraz na racjonalne określenie treści i zakresu elementarnego wykładu informatyki. Program takiego wykładu nie może ograniczać się do wyjaśnienia zależności matematycznych oraz technicznych, lecz winien być powiązany z jednej strony z rzeczywistością społeczną, z drugiej zaś — z praktyką wdrażania systemów informatycznych. Dopiero taki program stanowić będzie pełną, zamkniętą całość dostosowaną do konkretnych potrzeb społecznych.

UWAGA OŚRODKI OBLICZENIOWE

Biuro techniczne firmy CONTROL DATA CORPORATION w Warszawie oferuje do bezpośredniej sprzedaży za złote, fabrycznie nowe wzorcowe pakiety dyskowe przeznaczone do konserwacji technicznej (CUSTOMER ENGINEERING DISK PACKS), a mianowicie:

- dla jednostek pamięci dyskowej dostosowanych do wymiennych pakietów 6-płytowych w cenie 29.000 zł,
- dla jednostek pamięci dyskowej dostosowanych do wymiennych pakietów 11-płytowych w cenie 42.000 zł.

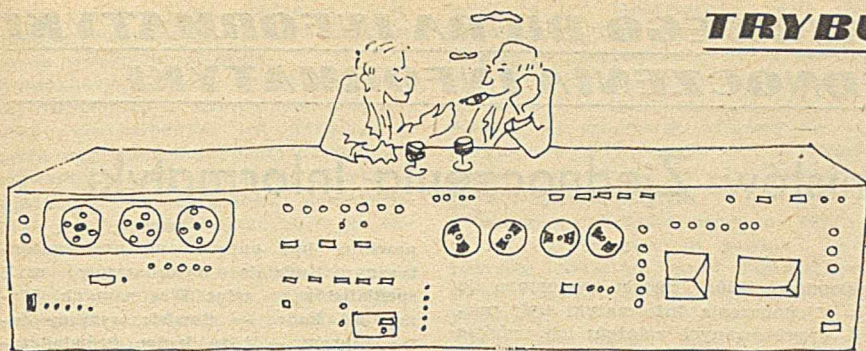
Powyższe pakiety wzorcowe stosowane są do kontroli i regulacji podstawowych zespołów zapisu/odczytu w jednostkach pamięci dyskowej (zwłaszcza do ustawienia głowic oraz mechanizmów wybierania ścieżek). Oferowane pakiety

charakteryzują się uniwersalnością rozwiązań i dlatego mogą być używane nie tylko do kontroli i regulacji jednostek produkcji firmy CDC, ale również innych producentów.

Warto podkreślić, że wzorcowe pakiety dyskowe są podstawowym powszechnie używanym składnikiem wyposażenia kontrolnego zespołów konserwacji technicznej pamięci dyskowych. Pakiety te gwarantują spełnienie tak istotnego warunku eksploatacji systemu komputerowego, jakim jest zapewnienie pełnej wymienności zapisanych na dyskach informacji pomiędzy różnymi jednostkami.

Ośrodki obliczeniowe zainteresowane nabyciem wymienionych pakietów wzorcowych mogą zwracać się w tej sprawie bezpośrednio do:

CONTROL DATA CORPORATION
BIURO TECHNICZNE
00-020 Warszawa, ul. Szpitalna 1/44
tel. 27-63-11 telex 81-38-18



Metody stosowania programów etykietujących

Gospodarka taśmami magnetycznymi, zarówno ich eksploatacja jak i przechowywanie, jest niejednakowo zorganizowana w poszczególnych ośrodkach obliczeniowych. Mimo tej różnorodności wszędzie nadaje się taśmom magnetycznym kolejne numery, które są uwidocznione na zewnętrznej części szpuli i jednocześnie zapisuje się je na TM z chwilą etykietowania. Dzięki tej numeracji istnieje możliwość określenia fizycznej tożsamości danej TM, co jest niezmiernie ważne w zakładaniu i gospodarce bankiem danych. Problem jest jednak w tym, że po pewnym czasie „giną” zapisane w etykiecie numery, a na ich miejsce pojawia się nr 00000012, lub 00000000.

Na podstawie długich doświadczeń i wykonania ogromnej liczby prób można stwierdzić, że przyczyną powyższego zjawiska jest nieumiejętne stosowanie programów etykietujących oraz brak szczegółowych instrukcji operatorskich.

ODRA 1304 ma trzy standardowe programy etykietujące:

— program # XQMY # ELIB
 — program # XQMV # TAPE
 — program # QZMY # TAPE

Program # XQMY # ELIB nadaje się do etykietowania taśm „dziewiczych”, to znaczy takich, które po pobraniu z magazynu mają być włączone do programów eksploatacyjnych. Program ten nadaje taśmie magnetycznej żadaną etykietę lecz do komórki zapisującej w etykiecie numer TM, wpisuje same zera. Fakt ten spowoduje rozbieżność z zaprowadzoną ewidencją taśmoteki i nieporozumienie w toku eksploatacji programów, jeśli wszystkie taśmy magnetyczne będą posiadały ten sam numer, czyli zera. Ponadto ośrodki, które stosują programy standardowe FIND i DMS nie będą mogły ich eksploatować, gdyż wymagają one konkretnych numerów w etykiecie.

Program # XQMV # TAPE nadaje się do etykietowania zarówno taśm dziewiczych jak i taśm roboczych, to znaczy TM, którym już nadano etykietę, lecz jest kłopotliwy w eksploatacji. Ażeby nadać odpowiedni numer TM, program wymaga wyperforowania karty parametrycznej lub taśmy papierowej, co łączy się

z niepotrzebnym wydłużeniem czasu i przestojami maszyny cyfrowej.

Program # XQMY # TAPE może być stosowany tak jak program # XQMV. Wadą jego jest to, że przy etykietowaniu taśm roboczych działa bardzo wolno (cztery odwinięcia taśmy), natomiast przy etykietowaniu taśm dziewiczych wymaga perforacji parametrów, tak jak program poprzedni.

W tej sytuacji opracowany został dodatkowo program etykietujący # ZQMY, posiadający następujące możliwości:

— wejście 20 — odczytanie etykiety
 — „ 21 — zapis SCRATCH TAPE
 — „ 22 — nadanie etykiety
 — „ 23 — SKREZOWANIE TM
 — „ 28 — podanie daty zapisu
 — „ 29 — MS na perforator

Proszę mi wybaczyć to niepolskie słowo SKREZOWANIE, lecz tak nazwali autorzy # ZQMY daną funkcję programu. Zresztą takich makaronizmów w dziedzinie informatyki jest pełno.

Wracając do tematu należy stwierdzić, że wspomniany program ma połączone możliwości pierwszych trzech programów etykietujących. Wadą jego jest to, że nie posiada ścisłych instrukcji operatorskich a nieumiejętne stosowanie różnych wejść do programu może spowodować zapisanie w etykiecie TM cyfry = 00000012. Następstwa tego mogą być wprost nieobliczalne.

Proszę sobie wyobrazić, że nawet program standardowy FIND i DMS uzna każdą taśmę za dobrą, jeśli ma ona wymagany numer w etykiecie.

Ponieważ wszystkie TM będą miały numer 12 stąd mogą być potraktowane jako te same. Myślę, że o możliwych pomyłkach przy takiej jednolitej numeracji nie warto nawet wspominać.

Z wymienionych programów etykietujących najlepszy i najbardziej uniwersalny jest jednak # ZQMY. Wymaga jedynie jednolitej i umiejętniejszej eksploatacji. Ażeby więc dobrze i bezbłędnie można było pracować, aby praca była szybka

i niezawodna, a jednocześnie aby ewidencja TM była jednolita od chwili nadania jej odpowiedniego numeru, aż do ostatniej chwili jej eksploatacji, proponuję stosowanie następujących instrukcji operatorskich:

1. Etykietowanie TM — nowych
 GI # ZQMY XX
 AL # ZQMY O ZZZZ
 GO # ZQMY 21
2. Etykietowanie TM — roboczych bez odczytu
 GI # ZQMY XX
 GO # ZQMY 23
3. Etykietowanie TM — roboczych z odczytem
 GI # ZQMY XX
 GO # ZQMY 20
 GO # ZQMY 21
4. Zmiana etykiety
 GI # ZQMY XX
 AL # ZQMY O ZZZZ
 GO # ZQMY 22
 MS # ZQMY = „nazwa etykiety”
5. Odczyt etykiety
 GI # ZQMY XX
 GO # ZQMY 20

U w a g a:

- a) Nie wolno zmieniać kolejności wejść,
- b) Nie wolno stosować oddzielnie wejścia 21,
- c) Nie wolno łączyć wejścia 20 i 23,
- d) Każdorazowo sprawdzić zapisany numer z numerem na szpuli,
- e) XX — wpisać numer PT-2.
- f) ZZZZ — wpisać numer archiwalny TM.

Ścisłe stosowanie podanych przeze mnie instrukcji operatorskich, gwarantuje pozostawienie na stałe raz nadanego numeru taśmy magnetycznej i umożliwia stwierdzenie fizycznej tożsamości danej taśmy. Ponadto stosowanie wyłącznie jednego programu etykietującego, zamiast istniejących czterech, wprowadza uporządkowaną pracę operatorską w OPI.

Pragnę zwrócić jednak uwagę na ścisłe przestrzeganie podanych wyżej zasad przy stosowaniu # ZQMY. Dla przykładu podam, że połączenie wejścia 20 i 23 spowoduje zapisanie w etykiecie cyfry 12, gdyż są to wejścia wzajemnie się wykluczające.

Wejście 20 jest odczytaniem etykiety a wejście 23 jest nadaniem etykiety bez odczytu, stąd program mając dwie sprzeczne instrukcje spowoduje wprowadzenie do etykiety cyfry 12. Podobnie jest z pozostałymi ograniczeniami.

Jednocześnie pragnę zaapelować do wszystkich Kolegów informatyków, by dzielili się z nami na łamach INFORMATYKI podobnymi doświadczeniami. Będę równocześnie wdzięczny każdemu z Kolegów za uwagi względnie dalsze rozwinięcie omówionego przez mnie tematu.

Zygmunt Topolewski

I Sesja Programistów Zjednoczenia Informatyki

BIAŁOWIEŻA 1973

Nie było przypadkiem, że pierwszą grupą zawodową, która została zaproszona do przedyskutowania celów i programu rozwoju Zjednoczenia Informatyki byli programiści. Oni bowiem zdecydowali o realizacji wypracowanych założeń; ich twórcze i organizacyjne zaangażowanie oraz identyfikacja z celami ZI będzie gwarancją powodzenia. Deklarację, która uwieńczyła dyskusję zorganizowaną w grudniu 1973 r. w Białowieży — drukowaliśmy w nr 3/74 INFORMATYKI. Ujęła ona takie zagadnienia, jak: charakterystyka produktu wytwarzanego przez programistów, problemy socjalno-ekonomiczne środowiska zawodowego, zagadnienia szkolenia i doskonalenia kadry programowo-projektowej w sieci ZETO. Wobec dużego zainteresowania 2 dniowymi obradami programistów oraz wielu pytań dotyczących szczegółów sesji i poszczególnych wypowiedzi, ZI oraz Klub Programistów zdecydował opublikować fragmenty tej dyskusji, wprowadzając one nie tylko w problematykę ale w jej nastrój i temperaturę.

Z kilkusetstronicowego stenogramu dyskusji białowieżskiej wybraliśmy kilkanaście — naszym zdaniem — najciekawszych wypowiedzi, które może staną się podstawą dalszej dyskusji w środowisku informatyków.

RYSZARD TEREBUS — ZJEDNOCZENIE INFORMATYKI

W naszej sieci popełniliśmy bardzo dużo błędów, biorących się głównie z rozproszenia sił Zjednoczenia. Można tę sytuację złożyć na karb okresu uczenia się, zbierania doświadczeń, ale to nas nie zwalnia od myślenia o przyszłości.

Bardzo dobrze się stało, że tym razem spotkali się — nie jak dotychczas — dyrektorzy zakładów, by radzić o sprawach „strategicznych”, lecz programiści i projektanci.

Czego oczekujemy od tego pierwszego spotkania?

Po pierwsze: chcemy w Zjednoczeniu Informatyki podjąć działalność zmierzającą do przemysłowej produkcji oprogramowania — ta dziedzina jest w Polsce nieznana, nie opanowana ani organizacyjnie, ani metodologicznie. Na Zachodzie zresztą również.

Druga sprawa: chcemy koniecznie, może w sposób nie formalny, a merytoryczny zintegrować środowisko programistów: Na razie myślimy o tym bardzo egoistycznie, to znaczy myślimy o naszej sieci ZETO. Niemniej jednak w przyszłości nie chcemy się ograniczyć tylko do naszego Zjednoczenia, o czym niech świadczy fakt, że zapraszaliśmy tu także ludzi spoza Zjednoczenia, między innymi z ELWRO. Sądzę, że dobrze będzie jeśli pewne ustalenia, wytyczne, pewne przemyslenia zrobimy właśnie w tym gronie.

Nawiązując do tej drugiej sprawy sądzę, że chyba wymyślicie na obradach jakieś formy organizacyjne dla dalszej działalności swego „towarzystwa”, „radę”, czy „klubu”. Można bowiem tak, czy inaczej oceniać pracę różnych organizacji społecznych. Nie ulega jednak wątpliwości, że zorganizowanie ruchu zawodowego pomaga, a nie przeszkadza w działaniu gospodarczym.

I proszę państwa, jeszcze parę słów o naszej przyszłości. Zauważamy, że ludzie młodzi, którzy dzisiaj przychodzą do pracy, już nie tylko patrzą ile zarabiają, ale liczą się i inne sprawy: karier zawodowych, doskonalenia, kształ-

cenia. To ważne i cenne, a zarazem zobowiązujące. Tym bardziej, że czeka nas naprawdę trudne zadanie oprogramowania jednolitego systemu. Takich maszyn w sieci Zjednoczenia na koniec 1975 roku będziemy mieli co najmniej 10. Będą to maszyny różne, maszyna poligonowa R-20, maszyna R-30, R-50. Mamy już praktyczne przykłady, że pewne maszyny jednolitego systemu w pełni akceptują oprogramowanie IBM. Powołaliśmy specjalny zakład oprogramowania RIAD jako oddział ZOWAR-u, pod nazwą R-POLSYSTEM, jednakże nowa generacja maszyn niesie nowe jakości, a my nie umiemy rozpakowywać oprogramowania, nie dajemy rady pakietom i wolimy wobec tego programować samodzielnie, od początku do końca. To trzeba będzie przełamać. Bardzo bym się cieszył gdyby dzisiejsze spotkanie było początkiem tego przełomu w podejściu do zagadnień programowania.

ANDRZEJ ARANOWSKI — OBRI

Cele Zjednoczenia już są, nie ma natomiast wystarczających i dobrych metod ich realizacji. A właśnie metody

powinny być opracowane przy autentycznym udziale wykonawców, czyli specjalistów, a więc Was samych. Mamy już kadre — dowód: wszyscy tutaj zebrani — i to kadre doświadczoną. Brakuje nam jednak określonych warunków organizacyjnych, ekonomicznych oraz tak zwanej „atmosfery”. Spróbujmy zastanowić się jak przełamać bariery przeszkadzające nam w efektywnej, satysfakcjonującej pracy. Zacytuje kilka ustępów z dokumentu, który niedawno się ukazał i który, uważam, jest bardzo cennym osiągnięciem nowego kierownictwa Zjednoczenia.

Jest to książeczka zawierająca cele działalności Zjednoczenia Informatyki na lata 1973—80. Dokument ten jeszcze formalnie do Was nie dotarł, ale pozwólcie, że przedstawię tu wybór tych celów. Cytuję: „działalność w sferze oprogramowania zostaje uznana za zasadniczą i podstawową dziedzinę aktywności Zjednoczenia. Składa się na nią produkcja oprogramowania, doradztwo organizacyjne dla użytkowników i projektantów systemów, udział w rozwoju ogólnokrajowych systemów informacyjnych i informatycznych. Należy stworzyć warunki, które spowodują, że do Zjednoczenia będą napływali najzdolniejsi i najbardziej twórcy programiści, oraz wszystkie wartości upowszechnienia pakiety programów i programy”. Zmiany organizacyjne, jak również prognoza rozwoju kadry Zjednoczenia wskazują na to, że podchodzi się do tego w tym programie perspektywnie i realistycznie. Proponuje się między innymi rewolucyjne — jak na nasze warunki — funkcjonowanie „instytucji”. Rewolucja ta polega na próbie powoływania samodzielnych jednostek organizacyjnych, to znaczy grup ludzi, którzy skupiają się na konkretnym zadaniu i — opracowując to zadanie — tworzą jednostkę samodzielną pod względem finansowym i w jakimś sensie również prawnym. Jest to ewenement organizacyjny w naszej sytuacji. Sądzę, że to się uda na przykład drogą umiejscawiania tego typu grup w charakterze jednostek organizacyjnych w naszej strukturze formalnej.

Cytowane posunięcia organizacyjne w Zjednoczeniu stwarzają korzystną atmosferę, aczkolwiek uwikłanie Centrali Zjednoczenia w system działania biurokracji centralnej i terenowej uniemożliwia często sprawne upowszechnienie, a następnie realizację decyzji. Na przykład cytowany dokument o celach Zjednoczenia na lata 1973—75. Przypadek sprawił, że mamy dziś możliwość zapoznania się z jego fragmentem najbardziej nas interesującym. A proszę państwa, mamy rok 1974! Chodzi o określone spłaszczenie struktur i delegowanie uprawnień na rzecz ośrodków, określoną centralizację decyzji w priorytetowych dziedzinach działalności centralnej.



Nasza z kolei piętą achillesową jest pozioma koordynacja między ośrodkami, a właściwie jej brak. Koordynacja pozioma polega na bieżącej współpracy i wzajemnym dopingowaniu się ośrodków, ich integracji a także wprowadzeniu i instytucjonalnym zagwarantowaniu prawa do ryzyka i wzmoczenia inowacyjności poszczególnych grup programistycznych i projektanckich.

A wystarczy także uprzytomnić sobie, że jeśli w 1975 roku w samym Zjednoczeniu, przewiduje się zatrudnienie 1500 samych programistów, to już w 1980 roku będzie ich 8.000. W sumie grupy programistyczne, projektanckie i doradców organizacyjnych w roku 1975 liczyć będą 2000 osób, podczas gdy w 1980 — liczyć będą już 9200 osób.

Chciałbym więc zaproponować skoncentrowanie się w dyskusji na trzech zasadniczych pytaniach, które wydają się newralgiczne dla naszej sytuacji: jak spowodować, żeby programista czuł się fachowcem właśnie jako programista, pozbywając się tym samym rozlicznych kompleksów w stosunku do przedstawicieli innych zawodów, jak zapewnić odpowiedni do wysiłku system wynagrodzeń i bodźców niematerialnych. Jak stworzyć odpowiednie warunki organizacyjne do efektywnego uprawiania zawodu programisty?

JACEK KARPIŃSKI — POLITECHNIKA WARSZAWSKA

Dlaczego zabieram głos na sesji programistów? Jestem konstruktorem systemów, konstruktorem maszyn cyfrowych, ale ponieważ uważam, że maszyny nie można traktować tylko jako bardziej lub mniej sprawnego żelazstwa, lecz jako kompleks razem z oprogramowaniem, czuję się wy tłumaczony. Najważniejszą teraz dla nas sprawą jest to, żebyśmy mogli efektywnie korzystać z maszyn, wszystko jedno jakie one są. Jeżeli zakres ich możliwości jest zgodny z oczekiwaniami i spełnia wymagania — to są dobre.

Do pewnych zagadnień musi być maszyna duża, do innych bardzo mała. W laboratoriach amerykańskich i na uniwersytetach spotyka się po kilka różnych typów maszyn i nikogo to nie martwi. Każda z nich może mieć zupełnie inny język programowania wewnętrznego, inny „assembler”. Ale używa się wspólnie języków wyższych rzędów np. PL/I lub FORTRAN, albo

jeżeli to jest laboratorium, gdzie się projektuje układy elektryczne, mechaniczne, czy architekturę, to używa się tam specjalistycznych języków programowania. I to też nikogo nie dziwi. Dlatego też proponuję, żebyśmy się za bardzo nie przejmowali, że są różne języki wewnętrzne, czy różna struktura wewnętrzna maszyn.

Programistę obchodzi program. Wydaje mi się, że program powinien być towarem. Mam na myśli zaprogramowany sposób rozwiązywania danego problemu.

Towar ten należy szacować według jego wartości, a nie — jak to ma niekiedy miejsce, według pracochłonności. Towar ten musi być pełnowartościowy. Warunkiem zachowania wartości oprogramowania jest jego właściwa konserwacja. Podobnie, jak z samochodami. Większość krajów rozwiniętych motoryzacyjnie ma przyzwoite stacje obsługi i konserwowanie samochodów nie jest problemem. My dopiero wkraczamy w fazę motoryzacji i konserwacja jest właściwie na etapie ząbkowania, nie tylko zresztą jeśli chodzi o brak stacji obsługi, ale o sposób traktowania tej sprawy. Wydaje mi się, że podobnie podchodzimy do sprawy serwisu oprogramowania. Dość powiedzieć, że w ELWRO na tym odcinku pracuje zdaje się 6 czy 7 fachowców, zamiast kilkuset, a do niedawna nie mieli oni nawet maszyny, na której mogliby przeprowadzać serwis oprogramowania. Spotykamy się z postulatami żeby pisać pakiety programów na małe konfiguracje maszyn. Nie jest to słuszne. Należy pisać pakiety w sposób modularny, parametryzowany, tak ażeby mogły one być adaptowane na zadane konfiguracje. Wiadomo, że im większa konfiguracja, to znaczy im większą ma pamięć operacyjną, pamięć dyskową itd., tym sprawniej program działa. Nie byłoby więc słuszne ograniczanie sprawności, wydajności programu. Programy modularne parametryzowane do zadanej konfiguracji same dopasowują się tak, jak trzeba.

I jeszcze jeden problem — wydaje mi się zasadniczy. Jak wiadomo programy się dzieli na: podstawowe, które służą do pisania programów i obsługi systemu oraz programy użytkowe. Mamy więc programistów systemowych i programistów użytkowych. W dawnych czasach, kiedy programowanie było sztuką elitarną i wymagało prawie nad-

przyrodzonych zdolności, przynajmniej tak się w gazetach pisało, jeżeli konstruktor mostu miał do rozwiązania problem ze swojej dziedziny, to zgłaszał się do takiego programisty i tłumaczył mu, jak to się ten most buduje. Programista przez trzy miesiące uczył się sztuki konstruowania mostu. Oczywiście się nie nauczył, ale w końcu pisał przez pół roku program nie rozumiejąc w ogóle problemu. Po wielu trudach, konsultacjach i nieporozumieniach wychodził jakiś dziwolak, który służył do obliczania mostu.

Można i tak, ale w Stanach Zjednoczonych już we wczesnych latach 60-tych, programowania, a więc użytkowania maszyn cyfrowych uczono na wszystkich wyższych uczelniach, na wszystkich kierunkach. W połowie lat 60-tych te same kursy przechodzili technicy i uczniowie w szkołach średnich. Dziś korzystania z maszyn uczy się dzieci w szkołach i w przedszkolach. Tak samo — jak uczy się mówić, jak uczy się pisać i rysować, tak w tej chwili należy uczyć korzystania z komputera, który nędznie stanie się narzędziem równie popularnym, jak maszyna do pisania lub froterka. Każdy specjalista w swoim zakresie powinien stosować nowoczesne narzędzia. W wielu już krajach nie ma w zasadzie programistów piszących małe programy użytkowe. Programy te piszą sami sobie potrzebujący nie programiści. Trudno nazwać ich programistami, tak jak trudno nazwać pisarzem człowieka, który pisze list na maszynie do pisania. Jest rzeczą pilną u nas szybkie przejście na ten poziom korzystania ze środków informatyki.

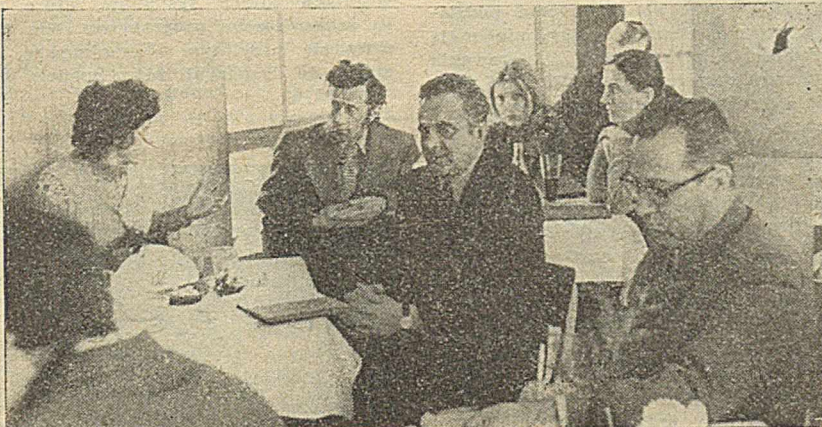
Jakie to pociągnie za sobą konsekwencje? Większość profesjonalnych programistów, profesjonalnych informatyków — bo ta profesja nie zanika — zmieni obszar swej pracy, przesunie się w kierunku pisania narzędzi do użytkowania systemów cyfrowych przez wszystkich.

Cała kadra profesjonalna programistów będzie robiła oprogramowanie podstawowe i wielkie pakiety parametryzowane, by każdy to mógł wziąć i u siebie zastosować. I nikt już nie przyjdzie do programisty i nie powie: bracie, ja mam taki problem, belkę muszę policzyć, bo każdy sobie sam policzy. Nie będzie, mam nadzieję, analfabetów informatycznych, jak to jest obecnie. Na szczęście w informatyce nie jesteśmy pierwsi na świecie i możemy korzystać z doświadczeń innych. Rzeczywiście popełnili oni sporo głupstw, mnóstwo ich to kosztowało, ale my spróbujmy pójść na skróty. Na przykład sprawa dużych programów. Dochodzimy obecnie do coraz bardziej skomplikowanych problemów, które wymagają bardzo dużych programów, całych systemów programowych. Taki jeden duży program na świecie ma ponad 200 tysięcy do miliona rozkazów. W Polsce nie pisaliśmy jeszcze takich dużych programów.

Otóż technika ich pisania musi być zupełnie inna.

Jak się to robi.

Wiadomo, że stosuje się w najszerszym znaczeniu modularność. Każdy pisać



cy nowy program usiłuje w miarę możliwości wykorzystać gotowe podprogramy czy kawałki programów, dostępne w bibliotekach danego systemu liczącego, danej rodziny maszyn, czy w danym języku. Ale muszą to być moduły napisane w sposób pozwalający na ich konsolidację.

Jest to sprawa, nad którą należy się merytorycznie zastanowić, i to nie tylko w ramach Zjednoczenia, ale w ramach kraju, a nawet w ramach naszego obozu.

Jak pisać programy dające się później łączyć w większe całości? Wbrew pozorom nie jest to sprawa prosta.

Przy konstruowaniu maszyn mówiło się dawniej o modularności w zakresie jednostki centralnej, w tej chwili mówi się, że tak należy konstruować cały sprzęt informatyki — nie tylko w ramach jednej rodziny — ale żeby można było dowolnie zestawiać te urządzenia w duże systemy, niezależnie od wytwórców. Tak samo tworzy się dziś modularne programy parametryzowane. Parametryzowanie jest sprawą znaną od lat przynajmniej kilku i stwarza realne możliwości produkcji przemysłowej. Ale niezależnie od przemysłowej produkcji oprogramowania, co stoi u nas na przeszkodzie w wykorzystaniu modułów pisanych przez kolegów, pisanych przez inne ośrodki, przez różne instytucje w kraju? W dużym stopniu przeszkodę tę stanowi prawo własności, prawo ich wykorzystywania.

Aby zdobyć dodatkowe środki płatnicze czyli bilety Banku Narodowego, każdy chowa swoje programy dla siebie.

Zjawisko to dało się obserwować na całym świecie. Stany Zjednoczone przeszły przez te wszystkie bolączki, które nas w tej chwili gnębią jakieś 10 lat temu i na dobrą sprawę do dziś się z nimi nie uporali. Problemy te są rozwiązywane częściowo przez organizacje użytkowników, każdy system ma swoją organizację n.p. IBM—Share, tak jak u nas Kluby Użytkowników. Sprawę prawa własności można rozwiązać nie poprzez patentowanie, nakazy, tylko w ten sposób, że raz przywoicie się płaci za wykonany program i programista traci do niego prawo.

Prawo własności a więc i prawo dysponowania programem należy do firmy, do Zjednoczenia, czy Klubu Użytkowników, w każdym razie przestaje ten program być obiektem sprzedaży indywidualnej.

Będzie to możliwe tylko wtedy, kiedy środowisko informatyków rzeczywiście będzie zintegrowane, kiedy będziemy się wszyscy znali i szanowali, wzajemnie obserwowali, ale z życzliwością a nie z zawiścią.

W tym celu dobrze byłoby zorganizować się, nie wiem czy to w postaci stowarzyszenia czy klubu ludzi, którzy naprawdę pracują w informatyce, którzy ją tworzą.

KRZYSZTOF JAROSŁAWSKI — POLITECHNIKA WARSZAWSKA

W okresie, kiedy pojawiły się pierwsze komputery, zainteresowali się nimi specjaliści służb finansowo-księgowych. Z tego też względu nawarstwiło się od



wielu lat ich zastosowanie w postaci różnorodnych systemów ewidencyjnych. Jest to efekt bardzo złego przykładu, który zacerpnęliśmy z gospodarki kapitalistycznej, pogłębiany przez zupełnie inne warunki i zupełnie inne zasady gospodarze.

Nie można więc dzisiaj mówić o efektach wykorzystania komputerów w gospodarce narodowej w takim stopniu, w jakim należało oczekiwać. Tutaj ponieśliśmy fiasko. Na zachodzie też zresztą mówi się o komputeryzacji, że nie spełnia pokładanych w niej nadziei.

Uważam, że w naszej gospodarce można i powinno się zastosować maszyny dla celów podejmowania optymalnych decyzji, zarówno w skali przedsiębiorstw, jak zjednoczeń i resortów. Brakuje nam jednak sprzętu i nie mamy określonego aparatu matematycznego, który umożliwiłby podjęcie zdecydowanych kroków w tym zakresie.

Drogi do takich rozwiązań szukać należy w metodzie programów parametryzowanych i programów modułowych. Ale takich programów nie można już robić w małych zespołach dwu, trzy osobowych. Jest nas tutaj prawie 80 osób, a to też za mało, żeby zbudować jeden przyzwoity pakiet.

Wicie, jak trudno jest skoordynować jeden program robiony przez trzech czy czterech programistów. Trudność ta wynika z tego, że kierownicy tematów nie mają opracowanych metod podziału tematu i zlecenia pracy, nie mają wypracowanej techniki kierowania większym zespołem pracującym w dziedzinie programowania nad jednym tematem.

I tutaj postulat pod adresem Dyrekcji Zjednoczenia: moim zdaniem zachodzi już konieczność podjęcia badań nad opracowaniem metodologii budowy dużych programów.

Następna sprawa. Wszyscy wiemy z naszej praktyki, jak łatwo zrobić program i jak trudno go pisać, a przecież szkoda wysokiej klasy fachowców do robienia dokumentacji. Wydaje mi się, że powinniśmy powołać nowy zawód: dokumentalisty programów.

Padło w dzisiejszych wypowiedziach zdanie o samopoczuciu programisty, że jest nienajlepsze, że cierpi on na coś w rodzaju kompleksu niższości. Informatyka i programiści zawsze będą pełnili rolę usługowe dla gospodarki narodowej, to nie ulega wątpliwości. Natomiast absolutnie nie mogą się zgodzić, że programiści czują kompleks niższości. Ta grupa ludzi czuje się dosyć mocno i wcale nie odczuwa kompleksów. Jest to grupa pełna inicjatywy, w wielu wypadkach otwierająca nowe drogi w strukturze zarządzania i organizacji, co jest wcale nietławe wobec częstych zmian w tej strukturze.

Systemy organizacji są tak szalenie zmienne w czasie, że program opracowany techniką tradycyjną jest nieaktualny już w momencie jego zakończenia. Tradycyjne techniki programowania nie dają już rezultatów. Musimy więc zastanowić się nad technikami pozwalającymi na opracowywanie takich pakietów, które uczynią pakiety (programy) nieczułym na zmiany organizacyjne.

Mam nadzieję, że w toku dzisiejszych obrad będą przedyskutowane sprawy związane z określeniem miejsca programisty w gospodarce narodowej, jego zadań i rangi zawodowej. Czy programista to zawód uznany przez inne grupy zawodowe, czy też jesteśmy w dalszym ciągu chałupnikami? Jeśli odpowiedź jest pozytywna, podnosząca programistę do rangi zawodu, to należałoby podnieść też sprawę uznania tego faktu przez czynniki oficjalne i przyznania programistom praw z tytułu zawodu wynikających.

STANISŁAW MROZIK — ZOWAR

Jak można wykorzystać w sposób bardziej efektywny cały potencjał, jaki w zakresie programowania reprezentuje Zjednoczenie Informatyki?

Wydaje mi się, że są tylko dwie możliwości: albo Zjednoczenie będzie jednostką niezwykle silną merytorycznie i wykorzysta całą sieć do realizacji prac według z góry określonych i ściśle koordynowanych koncepcji, albo trzeba stworzyć mechanizm, który samoczynnie sprawi, że prace pójdą we właściwym kierunku.

Chodzi tu po prostu o stworzenie silnej grupy pracowników koncepcyjnych, którzy robiliby określone projekty i kontrolowali ich realizację do poziomu najniższego.

Z punktu widzenia programowania wystarczy, że nie będziemy dopuszczać do instalacji takich systemów, które nie są zabezpieczone programowo. Nie musimy się przy tym przejmować, że mamy do dyspozycji różne modele ma-



zyn. Trzeba tylko ustalić obowiązująco języki programowania i przystąpić do instalowania dowolnego sprzętu. To jest sprawa kluczowa.

Natomiast jeżeli chodzi o sprawy ekonomiczne, to oczywiście, program musi być towarem, ale częściej występuje jego zastosowanie. Trzeba więc sprzedawać i zastosowania. De facto my w tej chwili nie sprzedajemy nawet programu tylko jego pracochłonność, czyli bardzo nieprzekonywającą podstawę ceny programu. A powinieliśmy działać taki mechanizm ekonomiczny, który spowoduje, że im częściej stosujemy nasz program, tym większe będziemy mieli z niego wpływy. Ktoś sobie może kupić program i mieć go do wyłącznej dyspozycji, płacąc odpowiednio więcej a ktoś inny może kupować tylko jego zastosowanie za odpowiednio mniejszą sumę, ale za to częściej. Program będzie stanowił dla kupującego cegiełkę w budowaniu na własny użytek oprogramowania, a zastosowanie posłuży do bezpośredniego użytku, bez możliwości konstruowania odrębnej całości.

Takie potraktowanie programów i ich zastosowań wymaga powołania twórczych zespołów projektowo-programowych. Sprzedawanie zastosowań i oprogramowania jest więc pewnym rozwiązaniem o tyle, o ile zespoły takie będą miały ze swej pracy bezpośrednie korzyści. Bo jeśli to będzie tylko wpływ dla instytucji, albo jak dotychczas nie będzie żadnego wpływu, to oczywiście pod znakiem zapytania będzie stała ekonomiczność całej tej działalności. Skoro już mowa o korzyściach materialnych to wydaje mi się, że programiści powinni dostawać dodatek za znajomość języków programowania. Wiem, że kryteria przyznawania takich dodatków to sprawa skomplikowana. Bez sensu byłoby np. dawać dodatek inżynierom za znajomość języków do opisywania określonych problemów. Zgłaszam jednak wniosek, ażeby rozpatrzyć ten problem. Zgłaszałbym również postulat zmiany dotychczasowego systemu wynagrodzeń. Nie chodzi o to, żeby zwiększyć płace, tylko o to, żeby płacić za wyniki pracy. Czyli innymi słowami, aby istniał taki system, który preferowałby ludzi zdolnych i dobrze pracujących. Jeżeli będziemy płacili — jak dotychczas — tylko za czas pracy, to nigdy nie będziemy mieli dobrych i szybkich wyników.

ROMAN JELIŃSKI — ZETO GDYNIA

Wydaje się, że w ZETO — Gdynia zrobiliśmy pewien krok w kierunku przemysłowej produkcji programów.

Przed wszystkim chciałbym podkreślić dwa elementy: pierwszy to narzędzia, którymi wykonujemy oprogramowanie, a drugi — to organizacja procesu wykorzystania istniejącego oprogramowania.

Stosowaliśmy sprzęt ICL 1900 i jego standardowe oprogramowanie. Jak wyglądała organizacja procesu programowania? Został on podzielony na dwa etapy: pierwszy można nazwać projektem oprogramowania, drugi — wytworzeniem kodu. Zastosowaliśmy podział na moduły, na tak zwane segmenty. Projekt zaczynał się wejściowymi założeniami do tego problemu, a kończył się dokumentacją.

Od sposobu prowadzenia dokumentacji może zależeć efektywność samego programowania. Dokumentacja ta zawierała opis algorytmów w języku programu, czyli tu oddzieliłmy opis algorytmu języka programowania. Później był projekt segmentów, który zawierał nazwę dokonywanego algorytmu i sposób łączności z innymi segmentami. Po utworzeniu tej dokumentacji przystąpiono do kodowania w oparciu o opis modułów. Programista, a właściwie człowiek, który kodował, otrzymał opis modułu, który był częścią dokumentacji. Od tego momentu rozpoczyna się drugi etap, proces wytworzenia kodu. Chciałem podkreślić jeszcze raz szczególną rolę dokumentacji, ponieważ w oparciu o dokumentację można użyć oprogramowania firmowego do automatycznego wygenerowania segmentów. Ale warunek: dokumentacja musi być precyzyjna i musi być precyzyjny opis projektu.

I jeszcze jedna sprawa. Nie można żądać od programistów integracji tematyki. Wydaje mi się na przykład niemożliwe zintegrowanie całej materiałówki w Polsce. Wśród programistów moim zdaniem powinno się wydzielić dwie grupy ludzi; jedna grupa tych, którzy tworzą narzędzia do tworzenia oprogramowania i grupa takich programistów, którzy tworzą nowe oprogramowanie użytkowe.

ANDRZEJ RAMUŁT — ZETO WROCŁAW

Chciałem przedstawić powstałą w ZETO-Wrocław koncepcję, jeszcze nie całkiem skryształizowaną, na temat przemysłowej produkcji oprogramowania dla maszyn serii ODRA 1300.

Jest w tej chwili zainstalowanych w Polsce 97 maszyn ODRA-1300, z tym, że prawdopodobnie będzie ich jeszcze około 200. Pracować będą mniej więcej do roku 1980, a nawet może i dłużej. W związku z tym uważamy, że już najwyższy czas zająć się sprawą ich racjonalnego i efektywnego oprogramowania. Maszyny ODRA 1300 są wykorzystywane już wiele lat, ale ich oprogramowanie firmowe jest jeszcze nierozpakowane, a tym samym słabo rozpowszechnione. Słaba jest też nadzieja na jakąś poprawę przy takiej, jak do-

tychczas, organizacji działania. Przed wszystkim zbyt mało ludzi się tym zajmuje.

Następna sprawa, to produkcja oprogramowania standardowego. Jest to związane z przeprowadzeniem pewnego rozeznania, jakiego rodzaju oprogramowanie jest potrzebne. Myślimy o oprogramowaniu typowych procesów technologii z zakresu EPD, czyli uzupełnienia istniejącego oprogramowania firmowego oraz o oprogramowaniu pewnych urządzeń zewnętrznych, w które wyposażone zostały maszyny ODRA 1300. Te prace częściowo są w toku, prawdopodobnie nie tylko w ZETO-Wrocław, ale i w innych ośrodkach. Wydaje mi się jednak, że ich cykl jest zbyt długi, a współpraca daleka od doskonałości.

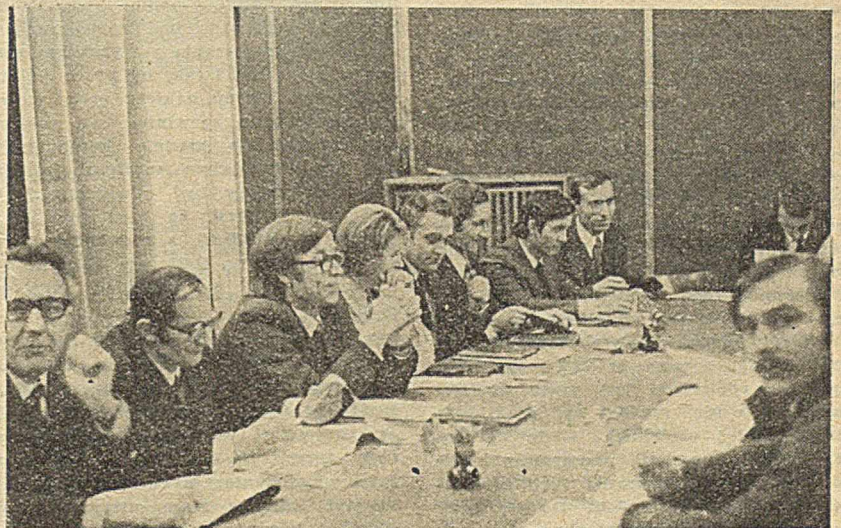
Sprawa serwisu oprogramowania. W tej chwili mogłoby to dotyczyć tylko ZETO-Wrocław, jednakże w przyszłości można problem rozszerzyć także na inne nasze ośrodki, albo na zasadzie dobrowolności, względnie na zasadzie wprowadzenia i koordynowania prac ogólnie przez Zjednoczenie, co gwarantowałyby szybkość i jednolity standard opracowań. Nasze rozeznanie wskazuje, że wiele Zakładów chętnie by z nami współpracowało.

Potrzebna nam jest jednak koordynacja działań, na przykład w postaci banku informacji, jak gdyby kartoteki programów gotowych oraz kartoteki programów, które są w trakcie opracowywania.

Te jednostki organizacyjne, które by przystąpiły do współpracy musiałyby przed napisaniem każdego programu kontaktować się z odpowiednią komórką organizacyjną aby uzyskać informację, czy przypadkiem taki lub podobny program nie jest już gdzieś opracowywany. Wiem, że istniały próby stworzenia takiej kartoteki w różnych instytucjach, ale jak do tej pory nikt się tym za bardzo nie przejmował, choć jest to sprawa wręcz paląca, jeśli chcemy myśleć o przemysłowej produkcji oprogramowania.

LESŁAW WOLAŃSKI — ZETO WROCŁAW

Pragnę przedstawić propozycję dotyczącą organizacji procesu produkcji serwisu systemowego. Systemowego, to zna-



czy obejmującego cały proces od marketingu poczynając, przez produkcję oprogramowania, do dystrybucji gotowych opracowań. Autorem omawianej propozycji jest dyrektor Trybalski z ZETO-Wrocław.

W każdym Ośrodku są służby projektowania i służby programowania. Służby projektowania dzielą się na analityków, projektantów, problemistów, technologów i programistów. Nasza propozycja jest następująca: tworzymy kilka obszarów działania. Pierwszy — to obszar użytkownika. Kontaktuje się on z Ośrodkiem przez służbę marketingową, która zajmuje się pełną obsługą użytkownika, poczynając od jego pierwszego wejścia do Ośrodka i przez wszystkie etapy, z którymi klient ma do czynienia. Jednocześnie marketing będzie miał pomoc w służbie serwisu systemowego. Drugi obszar obejmuje projektantów — problemistów. Ci ludzie precyzują zdefiniowane przez użytkownika zadania i przekazują je do następnego obszaru.

Obszar ten jest obsługiwany przez projektantów-programistów, czyli technologów, znających doskonale języki programowania, a jednocześnie mających rozeznanie w sprawach systemowych. Będą oni rzeczywistymi twórcami systemu, będą tworzyli drogą montażu pakietu, bądź też będą stawiali zadania następnej grupie, którą nazwalibyśmy — powędzmy — Zakładem Produkcji Oprogramowania. Zakład ten miałby dwa zasadnicze zadania: produkcję programów i rozpakowywanie oprogramowania opracowanego przez inne firmy, czy przez inne ośrodki.

ZDZISŁAW DĄBROWSKI — ZETO ŁÓDŹ

Trudno nie zgodzić się z oceną, że w naszej pracy myślimy jeszcze kategoriami drugiej generacji, że kwitnie partykularyzm, a rozwiązywania noszą piętno indywidualizmu, co bardzo utrudnia ujęcie ich w jakieś uniwersalne całości. Wydaje mi się, że jedną z przyczyn takiego stanu rzeczy jest zalamanie profilu działalności naszego Zjednoczenia; profil czysto usługowy, produkcyjny dominuje nad profilem szkoleniowym, choć ten ostatni determinuje postęp, a tym samym skuteczną działalność usługową na dziś, a przede wszystkim — na jutro.

Oto przykład: 70% czasu programisty zawarowane jest do sprzedaży. Jest to współczynnik nie stosowany nawet do obrabiarek. Wobec tego programista dąży do opanowania niezbędnego minimum kwalifikacji i wykorzystuje je do szybkiego produkowania nawet nieudolnych opracowań. Czy nie należałoby poprzez pewne zmniejszenie tych rygorów znaleźć czas na naukę? Oczywiście ten czas musi być ściśle zaplanowany, a kierunek tego szkolenia dokładnie określony. Powinien również powstać system sprawdzenia jego skuteczności.

Teraz druga sprawa: w ZETO typowym sposobem działania jest praca na zlecenie. Czy nie można by przejść — przynajmniej częściowo — do opracowywania i tworzenia pewnych produktów, a potem oferowania tych produktów i sprzedawania ich? Do tej pory robimy to odwrotnie, musimy się nagi-

nać do mało uniwersalnej specyfiki użytkownika. Tworząc coś wcześniej, dysponując uniwersalnymi opracowaniami, możemy przekonać użytkownika, że to on musi się dostosować do nas, na przykład doskonaląc swoją organizację.

IRENA MALERCZYK-DAŃDA — OBRI

Zarysowują się dwie strategie rozwiązywania problemu banku danych. Jedną z sugerowanych strategii, jest strategia wprowadzania w ów bank ewidencji programów gotowych i progra-



mów w trakcie opracowania. Jest to strategia doganiania.

Jest również sugestia podjęcia kierunku wyprzedzającego opracowywania przyszłych potrzeb. Dopiero znajomość potrzeb pozwoli sterować pracami nad oprogramowaniem. Byłabym za tą drugą strategią, przy czym w banku danych powinny się też znaleźć informacje o niedostępnych nam jeszcze środkach, czy technikach, stosowanych w krajach nas wyprzedzających. Proponuję zatrudnić futurologów i dokonywać prognoz na podstawie istniejącej analizy sytuacji — prognoz, dotyczących losu oprogramowania już za lat kilkanaście. Oprócz danych o produktach bank powinien być również wyposażony w dane o osobach, które ten produkt tworzą, z uwzględnieniem ich predyspozycji, zainteresowań i kwalifikacji.

STANISŁAW WÓJS — ZETO KRAKÓW

Jeżeli chodzi o kwestię oprogramowania, widzę potrzebę silnej koncentracji i koordynacji prac nad rozpakowywaniem oprogramowania z wyraźnym precyzowaniem cen i tak, by prace się nie dublowały. Pewne pakiety programowe, czy systemowe powinny być robione pod najmniejsze modele komputerów, pod ich najmniejsze konfiguracje, aby programy mogły być powielane. Może dojść do sytuacji, że ktoś robi jakiś system na RIAD-50, a nie będzie mógł go uruchomić na R-33, czy 45, jeśli będą one miały, na przykład, mniejszą pamięć.

Ponadto wydaje mi się, że ośrodki zetowskie bardzo dużo uwagi poświęcają oprogramowaniu obliczeń, przetwarzania systemów operowania danymi itd. Mało jest natomiast prac nad oprogramowaniem rozwiązań technicznych, matematycznych, inżynierskich. Moim zdaniem to jest jedna z szans utrzymania się ZETO i Zjednoczenia Informatyki na

pozycji wiodącej w informatyce krajowej, wiodącej w stosunku do ośrodków branżowych, które mają większe możliwości finansowe, dewizowe, niż Zjednoczenie Informatyki.

Mówiliśmy tu o formie dokumentowania programów, systemów, o mikrofilmach. Wszystko pięknie i słusznie. Tylko życzylibym sobie, żeby cała sieć ZETO stosowała jednolite oznaczenia graficzne, zgodne z polską normą. W związku z tym ktoś musi wyprodukować szablonny, wzory dokumentacji.

Następne pytanie: co jest z oprogramowaniem, i jak się zamierza wykorzystać maszyny MERA-302. Zjednoczenie Informatyki to kupiło i porzuciło do ośrodków. Ale jeżeli faktycznie Instytut Maszyn Matematycznych nie posiada oprogramowania, to czy dalej zamierza się go nam wyjechać. Nie ma sensu, by każdy ośrodek robił do tego oprogramowanie na własną rękę, może się co najwyżej na tym szkolić. Może by zebrać grupę osób z poszczególnych ośrodków i zrobić jakieś — przynajmniej podstawowe — oprogramowanie tego automatu obrachunkowego?

Uważałbym też jeszcze wyrazić nadzieję, że Zjednoczenie powoła zespół do opracowania poruszonych tu spraw i zaproponowania konkretnych rozwiązań do następnej dyskusji.

WŁODZIMIERZ PAŃKÓW — OBRI

Pragnę spojrzeć na informatykę i zawod programisty z perspektywy człowieka, który z technologią i techniką nie ma nic wspólnego, z perspektywy, z jakiej patrzy na te sprawy socjolog organizacji. Chciałbym wyłowić pewien wątek dyskusji i na tej podstawie sformułować problemy, które wiążą się z funkcjonowaniem zawodu programisty w Polsce.

Gdy się przyglądamy różnym zawodom w Polsce narzuca się generalny pogląd, że niezależnie, czy są to rolnicy czy chemicy, programiści czy też handlowcy, większość ludzi widzi w swych zawodach przede wszystkim to, co się nazywa techniką i technologią. Natomiast zachód już w latach sześćdziesiątych wyprzedził nas pod tym względem i skoncentrował się na tym, co by można nazwać ogólnie sposobami wykorzystania techniki, technologii i sposobami wykorzystywania ludzi do realizacji określonych celów. Wśród wypowiedzi, których miałem przyjemność dziś wysłuchać, przewija się już wątek oderwania od spraw elementarnych, jak technika i technologia, i próba przejścia do poszukiwania rozwiązań bardziej kompleksowych.

Ponieważ coraz częściej obserwuje się na świecie, że ludzie działają w sposób zorganizowany, postępują koncentracja działań w wielkich jednostkach organizacyjnych, a sprawą palącą staje się dostosowanie organizacji do wymiarów człowieka.

Przez wiele lat obserwowało się tendencje odwrotne, to znaczy usiłowanie włączyć człowieka w określone ramy organizacji i człowiek musiał się do nich dostosowywać. Od paru lat na zachodzie, a ostatnio i u nas, podejmuje się próby dostosowania organizacji do wy-

miarów człowieka i usiłuje się jakoś zgrać te dwa zasadnicze, nawet obiektywnie sprzeczne tendencje: z jednej strony wymagania stawiane człowiekowi przez organizację, z drugiej zaś — indywidualne dążenia ludzi działających w ramach organizacji.

Podstawowa kwestia, z którą by można się do państwa zwrócić, jest właśnie ta, jak widzicie, możliwość dostosowania naszych organizacji do ludzi w skali zarówno Ośrodków Obliczeniowych, jak i w skali całego Zjednoczenia. To ujęcie może brzmieć wyrotowo, ale trzeba na rzecz spojrzeć właśnie z tej strony. Bo nie o to chodzi, by Was przykrawać na miarę naszych organizacji, tylko wręcz przeciwnie, trzeba tak formułować nasze struktury organizacyjne, żebyście Wy w tych warunkach mogli uzyskać maksimum efektu i mieć poczucie satysfakcji z tego, co robicie. Wydaje się, że trzeba znaleźć sposób określenia Waszej roli organizacyjnej w ramach Ośrodków Obliczeniowych i szerzej.

A teraz już bardziej szczegółowe pytanie: jak w najbliższych latach ma wyglądać Wasz warsztat pracy, co trzeba zrobić, żeby ten warsztat dostosować do waszych potrzeb. Następna sprawa: jak Wy macie współpracować z innymi specjalistami. Bardzo ważny problem dotyczący przemysłowej produkcji oprogramowania — jakie musicie wypracować formy organizacyjne współpracy zespołowej. To jest zasadnicza sprawa, jeśli mamy zastąpić produkcję rzemieślniczą czy chałupniczą — produkcją przemysłową. Różnica ta będzie przede wszystkim polegała na koncentracji i — nieuchronnej w tych warunkach — dalszej specjalizacji w ramach Waszego zawodu.

Jakie mają być propozycje tej specjalizacji, żeby wydajność w Ośrodkach była najwyższa? Należy przedstawić w tej sprawie konkretne postulaty do działalności szkoleniowej i do doskonalenia kadry programistów.

Inna bardzo istotna sprawa to sprawa wyrysowania swoistych drzewek karier zawodowych, żeby uchronić specjalistów przed degeneracją w drodze awansu służbowego. Specjaliści muszą wyraźnie mieć przed sobą nakreśloną drogę, która by im starczyła na długie lata aktywnego życia, żeby nie dochodzili do przekonania, że co kilka lat znajdują się w ślepej uliczce, żeby nie musieli szukać innej formy działalności, szukać innych kwalifikacji.

W tym kontekście warto również wyraźnie domagać się sprecyzowania wymogów, które będą stawiane przy przejściach z jednego stanowiska na drugie; oczywiście ta drabina, czy to „drzewko kariery” musi być bardzo bogato rozgałęzione, żeby w każdym punkcie można było mieć przed sobą kilka alternatyw, ale równocześnie, żeby te cele były osiągnięte wyłącznie dzięki konkretnym sukcesom zawodowym, a nie dzięki temu, że się ma 6—8 czy 10 lat stażu.

Z tym łączy się system oceny pracy programisty; powinniście państwo jako osoby kompetentne, powiedzieć wreszcie, co Waszym zdaniem, powinno się brać pod uwagę przy ocenie Waszej pracy.

MIROSLAW GRELIK — OBRI

Jak zapewnić dopływ najzdolniejszych ludzi do Zjednoczenia Informatyki?

Chodzi mi o to, jak Waszym zdaniem, powinno funkcjonować Zjednoczenie, jakie warunki zawodowe i społeczne powinno zapewnić pracownikom. Powiedzmy IBM jest znane, że tam nikt nie zwalnia się z pracy. To jest tylko jedna z cech, która trzyma pracowników IBM. Jakie cechy powinno mieć Zjednoczenie Informatyki, żeby trzymało ludzi u siebie i żeby przyciągało innych? Mogą to być warunki płacowe, mieszkaniowe, zapewnienie interesującej pracy, czy wyjazdy zagranicę. Chciałbym się Was zapytać, które z tych czynników uważalibyście za istotne i najważniejsze?

Ludzie do nas przychodzą, czyli jakaś siła przyciągania w Zjednoczeniu Informatyki istnieje. Tym nie mniej obserwujemy dużą fluktuację kadr; w 1972 roku zwolniono 998 osób ogółem, a przyjęto 508. Co się za tym kryje?

Inny problem w Zjednoczeniu Informatyki. Programiści w wieku od 20 do 29 lat stanowią 73% zatrudnionych programistów, natomiast głównych specjalistów w tymże Zjednoczeniu Informatyki w wieku od 20 do 29 lat nie ma w ogóle. Wydaje mi się, że każdy programista, czy projektant, oprócz kariery czysto zawodowej, rozwoju własnego, jako projektanta i programisty, powinien mieć także perspektywę kariery, jako kierownika. Czy ją posiada? Fakt, że nie ma na wyższych i niższych stanowiskach kierowniczych osób z wykształceniem programistów jest niepokojący. Jeżeli nasze Zjednoczenie ma być organizacją dynamiczną, rozwojową, realizującą ambitne cele, o których tutaj mówiliśmy, to na pewno musimy młodszym stawiać odpowiedzialne zadania już wcześniej.



FRANCISZEK NIEMIEC — ZETO
WROCLAW

Nikt nie uwierzy, bo przecież jest wierutną bzdurą, że ośmiogodzinny dzień mieści w sobie 70 procent pracy efektywnej, a 30 procent to są szkolenia. Należałoby również usatysfakcjonować programistów czymś w rodzaju karty praw i obowiązków. Z tym się wiąże specjalny taryfikator i odpowiednie formy szkolenia, o których jeszcze powiem. Jeżeli chodzi o zabezpieczenie naszych interesów proponowałbym powołać jakiś związek zawodowy, stowarzyszenie lub klub zrzeszający programistów i projektantów.

Czy nie można zorganizować stałego ośrodka szkolenia i doskonalenia kadry Zjednoczenia? Byłby to jedyny ośrodek ze stałą kadrami i jednolitym programem. Może wtedy pozbylibyśmy się niejednoznaczności w nauczaniu.

Jeżeli chodzi o karierę zawodową, to zgadzam się, że u nas jest ona w ogóle źle pomyślana. Przykładowo powiem: lepszy programista, aby więcej zarobić, musi zostać kierownikiem, a wiadomo, że jeżeli zostanie kierownikiem, odsuwa się go od programowania, czyli że najlepszy pracownik przestaje być programistą, staje się administratorem. Czy nie można by struktury karier konstruować inaczej, tak jak to się zresztą dzieje na zachodzie, a ostatnio również na naszych wyższych uczelniach, gdzie są administratorzy i są fachowcy. Chodzi mi po prostu o to, żeby oceniać: dobry fachowiec, lepszy fachowiec itd. Można mówić o przeorganizowaniu dnia pracy. Nie wiem dlaczego mam pracować najwydajniej między 7 a 15. Czas pracy każdego pracownika jest podzielony na części — pracę indywidualną i koordynację z różnymi osobami. Więcej czasu powinniśmy poświęcać na pracę indywidualną.

MICHAŁ NOWICKI — ZETO
WROCLAW

Obawiam się, aby nie umknął nam problem ludzi, którzy są odpowiedzialni za to, żeby powstał program i żeby programiści dobrze pracowali. Myślę o kierownikach zespołów.

Byłem w Anglii na kursie, który zaczęto od omawiania tego, co kto może robić, jakie warunki musi spełnić, żeby był dobrym programistą, a jakie — żeby był, powiedzmy, dobrym kierownikiem itd. Są w tym zakresie wymiernenie wymagania. Kurs polegał tylko na dyskusji, a nie na wysłuchaniu przemówienia lektora. Swoją drogą, był to kurs poparty techniką na szczeblu kamery telewizyjnej i magnetowidu.

Cale szkolenie oparto na podręczniku planowania kariery ICL. Jest to taki poręcznik, w którym została rozpracowana ścieżka krytyczna pracownika, od stażysty do kierownika. Uczono nas również, jak człowieka motywować do pracy, co prawda uczono nas na zasadach zachodnich, niemniej pewne elementy z tego zawsze można zaadoptować, czego dowodem jest to, że ten kurs został powtórzony we Wrocławiu dla kadry kierowniczej wysokiego szczebla. Wspominam o tym dlatego, że widzę olbrzymią potrzebę takiego właśnie szkolenia, ale opracowanego i prowadzonego w Polsce. Dopiero po tym kursie uświadomiłem sobie, jak ważne są to sprawy dla sprawnej realizacji zadań. A więc: motywacja pracowników do pracy, czego się u nas nie zauważa i nie stosuje, chyba, że ktoś chce żeby mu ludzie dobrze pracowali i ma jakieś tam swoje amatorskie, intuicyjne sposoby. Musi mieć jednak co najmniej kilkuletni staż pracy, by wiedzieć, jak z człowiekiem postępować. Bo w motywacji jest i socjologia i psychologia, wszystkie nauki humanistyczne, których nas, jako matematyków, nigdy nie uczono.

Innym temat: stawianie celów. Mało kto u nas umie stawiać cele, wyznaczać programiście terminy: schematu blokowego, kodowania, testowania, dokumentacji. Tego typu cele powinno się stawiać z dokładną datą, z podkreśleniem celów głównych i szczegółowych. U nas nikt z kierowników nie jest obowiązany tego stosować, być może dlatego, że mało kto potrafi to robić. Sprawa komunikacji między ludźmi. Trzeba tak rozmawiać z programistą, żeby rozumiał, o co chodzi. Dano nam na wspomnianym kursie przykłady, że gdy się mówi do grupy ludzi, każdy inaczej to rozumie. Znakomite testy udowodniły, jak zawodna jest tu komunikacja. W związku z tym uczono nas metod postępowania, komunikowania się z ludźmi.

Jeszcze inna sprawa — rozpoznawanie problemów, z którymi borykają się pracownicy, usuwanie przeszkód, pomoc w rozwiązywaniu trudności. Kierownik musi umieć patrzeć na swoich podwładnych i widzieć ich w czasie pracy. A na jakiej zasadzie opiera się kontrola, na jakiej zasadzie ocena, a na jakiej zasadzie oszacowanie pracy programisty? Tego nikt u nas nie wie, a tym samym nie jest zobowiązany stosować. Ba, nawet było by niedobrze, gdybyśmy nagle zaczęli stosować, bo mało kto robi to dobrze. Trzeba więc szkolić ludzi w tak szerokim zakresie, jak to się robi na zachodzie. Niewielu może tam wyjechać, ale przecież są wzo-ry, próbowaliśmy — jak już wspomnia-łem — zrobić to we Wrocławiu.

RYSZARD TEREBUS — ZJEDNOCZENIE INFORMATYKI

Odpowiem na parę pytań.

Sprawa 70-procentowej wydajności produkcyjnej programistów — to sami dyrektorzy wymyślili. Powiem nawet dla-czego: my cały czas mówimy, że już nie możemy być ośrodkami obliczenio-

wymi, bo nasze usługi nie tylko spró-wadzają się do usług obliczeniowych. Chcemy je rozszerzać o sferę organiza-cji, zarządzania, doradztwa itd. Można by powiedzieć, że jesteśmy ośrodkami informatycznymi, ale mamy plan finan-sowy, który obowiązuje w czysto usłu-gowych ośrodkach obliczeniowych. Bie-rze się to i stąd, że liczba programi-stów w stosunku do ogółu zatrudnio-nych jest u nas za mała. Pożera nas aparat administracyjny, a ktoś przecież musi zarabiać na ośrodki świadcząc w jego imieniu usługi.

Sprawa fluktuacji kadr. To prawda, że my nie mamy najlepszych programi-stów, chociaż mamy bardzo dużo do-brych. Jest kilka przyczyn tego stanu rzeczy. Jedna z nich jest taka, że my historycznie najwcześniej powstałiśmy, wobec czego jesteśmy producentem kad-ry informatyków dla całej gospodar-ki. Musimy i będziemy musieli odda-wać ludzi na zewnątrz, gdzie zwykle czeka ich bardzo szybka kariera.

Znam osobiście kilku starszych progra-mistów, których szef zdrowo wygląda i na razie nie zamierza przenieść się do innego miasta; ich kariera jest dosyć problematyczna. Co więc robią? Idą na-tychmiast na szefa do nowo powstają-cego ośrodka. Drugi czynnik, sądzę, że najważniejszy, to są sprawy socjalne, mieszkaniowe. Ludzie odchodzą tam, gdzie dostaną mieszkanie, bo to prze-cież ważna sprawa. I jest jeszcze gru-pa — najinteligentniejszych programi-stów, którzy odchodzą, ponieważ nie bardzo jakoś widzą siebie w Ośrodkach Obliczeniowych typu usługowego. Wy-daje się, że ci ludzie odchodzą, bo nie mają po prostu ciekawej roboty, bar-dzo często nawet mniej zarabiają. Ale mnie się wydaje, że ta sprawa świadczy o pewnej nieudolności w ogóle nas wszystkich, kierownictwa Zjednoczenia także.

Za znajomość języków programowania nie będziemy dawali dodatków, to jest sprawa przesądzona.

Jestem przeciwnikiem fetyszyzowania stosunków panujących w informatyce

amerykańskiej. Rozmawiałem kiedyś z projektantami amerykańskimi, którzy powiedzieli — jesteście genialni, bo wy na MIŃSKach robicie takie systemy, jakich my byśmy nie potrafili zrobić na swoich IBM. Nie jesteśmy źli, nam brakuje doskonalenia i narzędzi. Cała rzecz polega na tym, że nie potrafimy pra-cować zespołowo, że mądrość nasza nie jest mądrością zespołową, a mądrością indywidualnych, wybijających się ludzi. To znaczy nie widzimy lasu, jako kom-pleksu, a widzimy trawę, krzak, drzew-ko, grzybek i żubra.

Program, jako towar. Jak się jedzie do Czechosłowacji na konferencję klubu użytkowników, tenże klub funduje do-bry hotel, dobre jedzenie i inne atrak-cje. Skąd to się bierze? Bierze się wła-snie ze sprzedaży programów. Oni się oparli na wzorcu stowarzyszeń progra-mistów amerykańskich, gdzie program jest odsprzedany za pewien procent, przy czym część pieniędzy, bardzo ma-ło, idzie do kieszeni programisty, a reszta idzie do kasy stowarzyszenia programistów. Stowarzyszenia programi-stów mają za to domy wypoczynkowe, stypendia zagraniczne i krajowe i tak dalej. Myśmy tej sprawy nie rozwią-zali, choć bojownikami o utowarwienie programów, o nowe prawo obliczeniowe jest Adam Empacher, który nota bene trochę się teraz zaniebwał w tych sprawach. Z wyjątkiem Polski i Związku Radzieckiego, to już nawet Bułgaria uważa, że program jest do sprzedania. Myślę, że sprawę rozwiążemy i musimy to zrobić szybko. Sprawa ta przeszkadza nie tylko w kraju, ale i we współ-pracy zagranicznej.

Nasze dzisiejsze spotkanie nosi numer 1, ale proponuję cykliczne spotkania tego typu. Nie mam w tej chwili zda-nia, czy raz, czy dwa razy w roku. Takie spotkania będziemy robili z udzia-łem przedstawicieli innych zawo-dów informatycznych, a nawet z dy-rektorami.

oprac. Ryszard Wiśniowski

O wpływie komputerów na społeczeństwo

Układ typowego artykułu na temat „Wpływ komputerów na społeczeństwo” wygląda następująco. Najpierw występuje grupa argumentów reprezentujących jedno stanowisko. Mówią one o tych wszystkich dobrodziejstwach, które maszyny przyniosły już społeczeństwu i często nawet usiłuje się dowieść, że bez „rewolucji komputerowej” porządek społeczny uległby rozkładowi. Potem następuje zwykle ostrzeżenie z przytoczeniem argumentów drugiego stanowiska, w którym mówi się o pewnych niekorzystnych zjawiskach powstających w związku z zastosowaniem komputerów. Wspomina się tu zazwyczaj o zagrożeniu strefy osobistej jednostek przez tworzenie banków danych personalnych oraz o niebezpieczeństwie znacznego wzrostu bezrobocia powodowanego przez rozwój automatyzacji.

Na koniec wreszcie pojawiają się peany na temat wspaniałej terażniejszości i przyszłych osiągnięć komputerów.

Dowodzi się też, że przytoczoną argumentację drugiego stanowiska, podnoszącego niebezpieczeństwa płynące z komputeryzacji, można w znacznej mierze pomniejszyć za pomocą różnych zabiegów technologicznych. Ustęp końcowy zawiera zwykle apel o szerokie poparcie przez społeczeństwo coraz to szerszych prac badawczych i konstrukcyjnych nad komputerami. Zwykle łączy się to z mniej lub bardziej subtelnym stwierdzeniem, że jedynie wiedza o maszynach, a więc jedynie naukowcy informatycy mogą uchronić świat od wyraźnie szkodliwych skutków komputeryzacji.

W rzeczywistości wpływ komputerów na społeczeństwo jest znacznie mniejszy niż ten, który sugerują nam środki masowego przekazu. Oczywiście są takie przedsięwzięcia, jak na przykład podróże kosmiczne, których bez maszyn cyfrowych nie można byłoby w ogóle zrealizować. Przemysł komputerowy, a wraz z nim szkolenie niezbędne do niespotykanej skali. Należy jednak stwierdzić, że przemysł ten przeważnie pracuje na własne potrzeby. Podobne jest to do ekonomiki funkcjonującej na wyspie, której mieszkańcy utrzymują się z prania sobie nawzajem bielizny.

Ta część, która pozostaje poza sferą owej samoobsługi jest w znacznym stopniu wspierana przez instytucje państwowe oraz inne gigantyczne przedsiębiorstwa, które „znają wartość wszystkiego z wy-

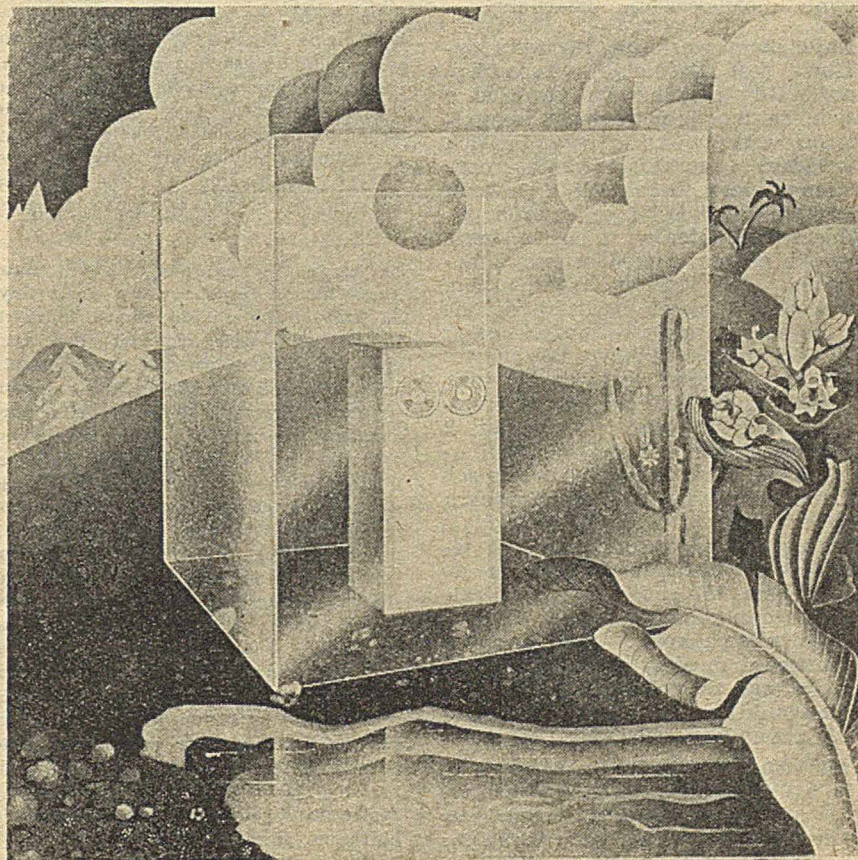
jątkiem ceny niczego”, tzn. znają bezpośrednią przydatność systemów informatycznych, a nie mają pojęcia o ich ostatecznym koszcie społecznym. Systemy rezerwacji biletów lotniczych czy skomputeryzowane szpitale obsługują jedynie drobną, lecz w większości przypadków najbardziej wpływową część społeczeństwa. Nie można więc na tej podstawie twierdzić, że systemy takie mają wpływ na całe społeczeństwo.

Skutki uboczne technologii

Istotnym powodem, dla którego odrzucam przedstawione tu przeze mnie w nieco karykaturalnej postaci argumenty jest to, że bezpośredni wpływ społeczny nowej, szybko rozwijającej się technologii jest niczym w porównaniu z jej o wiele subtelniejszymi i w ostatecznym efekcie ważniejszymi skutkami ubocznymi. W tym ostatnim aspekcie wpływ społeczny komputerów nie został jeszcze zbadany.

Aby lepiej zrozumieć znaczenie tych pośrednich, subtelnich wpływów technologii, rozważmy wpływ wynalazku mikroskopu na społeczeństwo. Gdy w

latach pięćdziesiątych XVII wieku wynaleziono mikroskop, panujące wówczas teorie uznawały, że choroba jest karą nałożoną na człowieka przez Boga. Sądzono, że ciało grzesznika jest nawiedzane przez tzw. humory, których równowaga zakłócana jest zgodnie z boską sprawiedliwością. Sposobu na wyleczenie choroby należało by więc poszukiwać przede wszystkim w pokucie, a następnie w próbie zrównoważenia humorów, na przykład przez upuszczenie krwi. Upuszczenie krwi było zarówno bolesne, a więc uwzględniające karę i pokutę, jak i potencjalnie równoważące, gdyż polegało na usuwaniu z ciała pewnej substancji. Mikroskop umożliwił człowiekowi zobaczenie mikroorganizmów i tym samym utorował drogę bakteriologicznej teorii chorób. Zdziwiająco odkrycie bardzo drobnych organizmów żywych zapoczątkowało również ideę ciągłego łańcucha życia, co z kolei było koniecznym, wstępnym warunkiem intelektualnym powstania teorii darwinizmu. Zarówno bakteriologiczna teoria chorób, jak i teoria ewolucji zmieniły zasadniczo wyobrażenia człowieka o Bogu a w konsekwencji i o sobie samym. W sferze polityki teorie te posłużyły

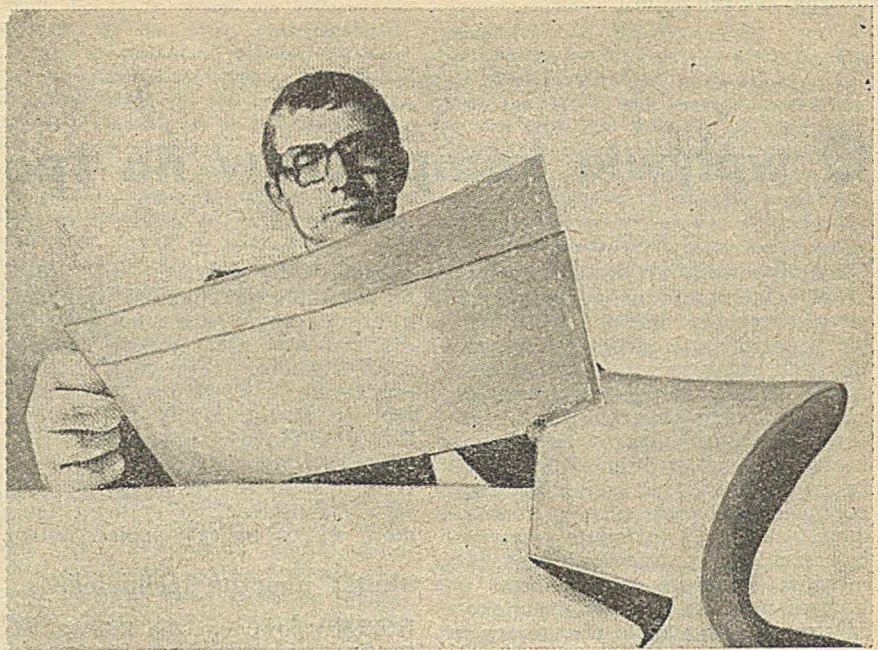


do zmniejszenia wpływów Kościoła oraz powszechnie uprawomocniły kwestionowanie podstaw niepodważalnego dotąd autorytetu. Nie twierdzą wcale, że wyłącznie mikroskop był odpowiedzialny za te szerokie przemiany społeczne, jakie nastąpiły po jego wynalezieniu. Uważam tylko, że wynalazek ten umożliwił takie przesunięcia we wzorcu myślenia, bez których zmiany te byłyby niemożliwe.

Powstaje więc uzasadnione pytanie, czy komputer spowoduje podobne zmiany w wyobrażeniach człowieka o sobie i czy zmiany te odpowiednio silnie wpłyną na społeczeństwo. Osobiście sędzę, że tak, chociaż muszę szybko dodać, że nie wierzę, aby maszyna powiedziała nam już dużo o człowieku i jego istocie. Aby zająć się bliżej tym zagadnieniem musimy zapytać, czym maszyna cyfrowa różni się od wielu innych maszyn zbudowanych przez człowieka. Człowiek zbudował dwa zasadniczo odrębne rodzaje maszyn: nieautonomiczne i autonomiczne. Autonomiczną jest taka maszyna, która działa przez długi czas nie na podstawie takich wejść ze świata zewnętrznego, jak na przykład czujniki czy czynności operatora, lecz na podstawie wprowadzonych do niej modeli pewnych aspektów rzeczywistego świata zewnętrznego. Przykładem maszyn autonomicznych są zegary, które działają na podstawie wprowadzonego do nich modelu naszego systemu planetarnego. Maszyna cyfrowa jest oczywiście przykładem idealnym. Można do niej wprowadzić modele o nieograniczonym w zasadzie stopniu złożoności i wierności, limitowanych jedynie możliwościami człowieka.

Największą wartością maszyny cyfrowej jest właśnie jej autonomiczność. Gdy mówimy na przykład o tym, że moc obliczeniowa komputerów wzrasta wraz z rozwojem ich urządzeń i oprogramowania, rozumiemy przez to fakt, iż z powodu wzrostu szybkości i pojemności pamięci komputerów, a może także dzięki nowym metodom programowania, do maszyn tych można wprowadzić coraz bardziej złożone i coraz wierniejsze modele coraz to większych odcinków rzeczywistości. Może okazać się tu co najmniej dziwnym, że podczas, gdy pokazujemy tak oczywistą autonomię w odniesieniu do maszyn, poważni uczeni w szanowanych uczelniach (mam tu na myśli D. F. Shinnera z Uniwersytetu Haward [1]) poddają w wątpliwość zagadnienie autonomii jako faktu w odniesieniu do człowieka. Nie sędzę, by pojawienie się tego paradoksu w tym miejscu było przypadkowe. Aby móc to zrozumieć musimy sobie zdać sprawę z tego, że poświęcanie się ludzi dla nauki zawierało zawsze w sobie czynnik masochizmu.

Już niejednokrotnie nauka doprowadzała nas do odkryć, które, gdy spojrzę na nie powierzchownie, pomniejszaly człowieka. Galileusz usunął człowieka ze środka świata, Darwin pozbawił go



odrębności od zwierząt, zaś Freud pokazał, że jego rozsądek jest złudzeniem. Człowiek jednak prowadzi swe dociekania dalej i głębiej. Nie mogą oprzeć się myśli, że istnieje analogia pomiędzy pogonią człowieka za wiedzą, a poddaniem się jednostki terapii psychoanalitycznej. Oba te działania są podejmowane przy świadomości tego, że odkrycia badającego mogą spowodować szkodę, jego wyobrażeniom o sobie samym; oba działania ukazują postanowienie raczej szukania znaczenia swej egzystencji poprzez walkę z prawdą bez względu na otrzymywane ciosy, niż godzenie się na życie bez znaczenia w świecie fałszywych złudzeń. Zdaję sobie jednak sprawę z tego, że czasami ludzie zaczynają psychoanalizę nie po to, aby zachować swe złudzenia i nie sięgać po rzeczywistość głębiej, lecz aby zamienić poglądy w siłę wpływów osobistych. Obserwacja ta nie podważa analogii do pogoni człowieka za wiedzą.

Za każdym razem, gdy odkrycie naukowe burzy dotychczasowe podwaliny, na których opiera się gmach ludzkich wyobrażeń o sobie, zachodzi silna reakcja taka, jak w przypadku zaistnienia podobnych okoliczności podczas terapii psychoanalitycznej. Rodzą się silne mechanizmy obronne, od zaprzeczania poczynając, a na racjonalizacji kończąc. I rzeczywiście, gdy pacjent zachowuje się tak, jak gdyby akceptował bez oporu jakiś burzący jego duszę pogląd, to psychoanalityk podejrzewa, że beztrojska pacjenta może z powodzeniem maskować negację przyznania temu pogładowi stanu prawdziwie operacyjnego w sferze wyobrażeń o sobie. Lecz co ma psychoanalityk sędzić o pacjencie, który pozytywnie przyjmuje zaferowaną mu — tytułem próby — kompletnie go upokarzającą wiedzę o sobie i natychmiast po jej przyjęciu przekształca ją w nowe podstawy swego życia? Taki przypadek jest z pewnością objawem poważnego kryzysu w życiu umysłowym pacjenta.

Sędzę, że obecnie jesteśmy świadkami początkowej fazy takiego kryzysu życia umysłowego naszej cywilizacji. Wspomniany przeze mnie mikroskop spowodował zmiany w autoportrecie człowieka. Nikt jednak nie mógł tego przewidzieć w połowie XVII wieku. Możliwość podobnego oddziaływania komputera wyraził w sposób niedościgniony jeden z moich szanownych kolegów — „Mózg jest po prostu maszyną z mięsa”. Myśl ta już teraz zaprzęta umysły akademików, inżynierów, przemysłowców i dziennikarzy. Jaki jest udział komputera w doprowadzeniu do tak smutnego stanu rzeczy? Należy od razu powiedzieć, że sam komputer nie jest głównym czynnikiem przyczynowym. Jest on jedynie skrajną ekstrapolacją technologii. Gdy traktujemy go, jako czynnik wywołujący pewien dogmat filozoficzny, stanowi on jedynie reduccio ad absurdum w stosunku do ideologii technologicznej. Ale jak można traktować go jako źródło dogmatu filozoficznego?

Teoria a praktyka

Należy sobie wyraźnie uświadomić fakt, że maszyna bez programu jest niczym. Program jest zasadniczo pewną transformacją jednej maszyny w inną, posiadającą autonomię i zdolność zachowywania się, w bardzo realnym tego słowa znaczeniu. Języki programowania opisują procesy dynamiczne i co jest bardzo istotne — opisywane procesy mogą być rzeczywistością realizowane. Tym samym możemy budować modele dowolnego aspektu świata rzeczywistego, aspektu, który nas interesuje i który możemy zrozumieć. Możemy również nasze modele uruchamiać. Należy jednak być uważnym i pamiętać, że model

maszynowy jest pewnym działającym opisem. Zwykle kiedy mówimy, że A jest modelem B, to rozumiemy, że teoria dotycząca niektórych aspektów zachowania się B stanowi również teorię tych samych aspektów zachowania się A. Z tego wynika, że gdy na przykład rozpatrujemy model maszynowy paranoi, podobny do opublikowanego przez Colby'ego i innych [2], to nie mamy podstaw sądzić, że model ten mówi nam cokolwiek o paranoi dlatego tylko, że odzwierciedla on w pewnym sensie zachowania się paranoika. Chociaż zwykła maszyna do pisania odzwierciedla w pewnym sensie zachowanie się nieśmiałego dziecka (na napisane pytanie nie otrzymuje się żadnej odpowiedzi), to jednak nie pomaga to nam w zrozumieniu tej nieśmiałości. Sprawdzenia modelu należy dokonać na podstawie jego teorii. Języki programowania dostarczyły naukowcom socjologom nowy oręż w tym sensie, że stosowanie ich pozwala uniknąć niejasności opisów dyskusyjnych. Nie zmniejszyło to jednak odpowiedzialności tych ludzi za tworzenie takich teorii, które mogą być obronione. Nawet błędy można wygłaszać z ogromnym formalizmem i elokwencją, ale przez to nie zamieniają się one w prawdę.

Niemożność dokonania rozróżnienia pomiędzy opisami, nawet tymi, które „działają”, a teoriami, wyjaśnia w znacznej mierze fakt, że ludzie odrzucający pogląd o człowieku, jako maszynie, zostali zepchnięci do defenzywy. Ostatnie osiągnięcia, dotyczące rozumienia przez maszynę języków przyrodniczych, dostarczają odpowiedniego przykładu. Halle i Chomsky, że wymienię jedynie tych dwu najlepiej mi znanych, długo pracowali nad taką teorią języka, którą musi spełniać każdy model wzoru językowego [3], [4]. Ich cel jest podobny do zamierzenia fizyka piszącego układ równań różniczkowych, które powinny być spełnione przez każdego rowerzystę. Żaden fizyk nie twierdzi, że aby stać się doskonałym rowerzystą, trzeba znać lub chociaż rozumieć rozwiązanie równań różniczkowych. Ani Halle, ani też Chomsky nie twierdzą, że ludzie znają lub świadomie przestrzegają reguł, które, jak sądzą, rządzą danym wzorem językowym. Halle i Chomsky, podobnie jak fizycy — teoretycy, dążą do zidentyfikowania z odpowiednimi elementami rzeczywistości, stałych i parametrów swych teorii. Stawiają oni hipotezę, że ich reguły tworzą rodzaj opisu rzutu pewne aspekty struktury umysłu ludzkiego. Ich problemem jest więc nie tylko odkrycie reguł ekonomicznych wyjaśniających wzór językowy, lecz również wywnioskowanie, jakie mechanizmy ekonomiczne określają dokładnie wybór odpowiednich reguł. Ponieważ są oni w ten sposób zmuszeni zwrócić się do umysłu ludzkiego, a nie tylko do mówiących językiem angielskim, muszą oni zajmować się z konieczności wszystkimi wzorami języków ludzkich, nie tylko języka angielskiego.

Bardzo znaczną część swej pracy poświęcili oni ilustracji obserwacji, że

we wszystkich językach ludzkich zdania deklaracyjne są często zamieniane na pytania poprzez zastawienie dwóch słów („Jan jest tu” — „Jest Jan tu?”). Opisanie reguł, które przekształcają zdania deklaracyjne w pytania jest jedną sprawą — prosta reguła permutacyjna jest tu oczywiście niewystarczająca. Drugą sprawą jest opis maszyny, która wymaga tych właśnie reguł, podczas gdy istnieją i inne, prostsze (przy jednakowych warunkach pozostałych). Dlaczego — na przykład — nie jest tak, że zdania deklaracyjne, czytane od końca, przekształcają się w pytania? Odpowiedź powinna być taka: inne ograniczenia istniejące w „maszynie” łączą się razem przeciwko tej wycinkowej prostocie na rzecz bardziej ogólnej ekonomiki mechanizmu. Przykłady takie ilustrują głębiej poziom wyjaśnienia, jaką próbują osiągnąć Halle i Chomsky. Nic dziwnego, że są oni przerażeni swoim problemem.

Ludzie pracujący nad programami rozumienia języków przyrodniczych operują w trybie nazywanym często trybem działania. Wygląda to tak, jak gdyby budowali oni maszyny, które mogą jeździć na rowerze stosując się do takich reguł empirycznych jak np. „jeśli czujesz przemieszczenie w lewo przenieś swój ciężar na lewo”. Może zachodzić (i często zachodzi) silna współzależność pomiędzy rozwijaniem teorii, a praktycznymi zadaniami konstrukcji systemów, których teoria nie jest jeszcze w pełni zrozumiała. Świadectwem tego jest ścisła współpraca pomiędzy aerodynamiką i konstrukcją samolotów w pierwszej ćwierci naszego stulecia. Tym jednak, co się liczy w trybie działania, jest nie wypracowanie teorii, lecz działanie systemów. I oto systemy wypracowane przez nową generację semantyków maszynowych zaczynają dopiero działać.

Od czasu uznania przez naukowców informatyków ważności powiązań pomiędzy syntaktyką i pragmatyką, a wraz z tym ważności przetwarzalnej przez maszynę wiedzy, zaczęli oni robić postępy. Być może pod koniec bieżącego dziesięciolecia pojawią się systemy maszynowe, za pomocą których będą mogli porozumiewać się w języku przyrodniczym tacy specjaliści, jak fizycy, chemicy i matematycy. Z pewnością pewna część tych osiągnięć będzie się opierała na sukcesach w innych dziedzinach, takich jak symulacja maszynowa procesów poznawczych. Jest rzeczą zrozumiałą, że każdy sukces w tej dziedzinie, nawet osiągnięty empirycznie, bez towarzyszącego mu wzbogacenia teoretycznego, może łatwo doprowadzić do pewnych złudzeń. Czyż nie jest łatwo spróbować uwierzyć, że maszyna, która rozumie język przyrodniczy, bez względu na to jak wąski jest jego kontekst, przejęła coś z istoty ludzkiej? Już Kartezjusz mógł w to uwierzyć. Istotnie, dzięki tej bardzo rozumiałej próbie, maszyna cyfrowa staje się źródłem dogmatu filozoficznego.

Kusi mnie by opowiedzieć, jak zbudowane są działające programy i jak, dzięki różnym kruczkom nie

pretendującym do posiadania jakichkolwiek podstaw teoretycznych, osiąga się poprawne działanie rzeczy, które dotychczas nie działały. Samo jednak pytanie — „Czy maszyna cyfrowa posiada coś z istoty człowieka?” — jest dygresją i w pewnym sensie pułapką. A to dlatego, że na podstawowe pytanie — „Czy człowiek rozumie istotę człowieka?” — nie może odpowiedzieć technologia, a więc z pewnością i żaden instrument technologiczny.

Metafora technologiczna

Pytałem tu wcześniej, co ma myśleć psycholog, gdy jego pacjent przyswoi sobie zaofertowaną mu celowo poniżającą interpretację i usiłuje natychmiast przekształcić ją w nową podstawę swego życia. Sądzę teraz, że sformułowałem to pytanie za słabo. Co by było, gdyby psychoanalitik jedynie chrząknął i to spowodowałyby wspomniane wyżej konsekwencje? Oto jest nasza dzisiejsza sytuacja. Nauka o komputerach, a w szczególności jej gałąź zajmująca się sztuczną inteligencją — chrząknęła. Być może, że prasa wzmocniła to chrząknięcie niewłaściwie, ale jest to mimo wszystko chrząknięcie. Nie mogę się powstrzymać od wyrażenia poglądu, że ta gotowość uwierzenia, iż to chrząknięcie odkryło nagle całą naturę człowieka, która okazała się mechanizmem zegarka, jest objawem czegoś straszliwego. Sądzę, że niesłusznie pozwoliliśmy takim metaforom technologicznym, jakie Mumford [5] nazywa „mitem maszyny” oraz samej technice, tak gruntownie przeniknąć w nasze procesy myślowe, że ostatecznie przekazaliśmy technologii nasz obowiązek formułowania pytań. Słusznie więc zauważają ludzie rozsądni, że duże banki danych i rozbudowana sieć maszyn cyfrowych zagrażają człowiekowi. Pozostawiają oni jednak technologii formułowanie odpowiednich pytań. Podczas gdy prosty człowiek zapytałby: „czy potrzebujemy tych rzeczy?”, technologia pyta: „jakie elektroniczne sztuczki uczynią je bezpiecznymi?”. Gdy prosty człowiek pyta: „czy to jest dobre?”, technologia zadaje pytanie: „czy to będzie działało?”. W ten sposób nauka, a nawet rozsądek stają się czymś, czym może operować technologia i większość maszyn cyfrowych. Nie należy tego uważać za przesadę. Cytuję tu wyjątek z pracy H. A. Simona, jednego z największych amerykańskich naukowców informatyków [6]:

„Jeśli uda się nam rozszerzyć i pogłębić naszą wiedzę — teoretyczną i praktyczną — o maszynach cyfrowych, odkrywamy fakt, że ich zachowaniem się rządzą w znacznej mierze prawa ogólne, że to, co wydawało się w programie maszyny bardzo skomplikowanym, było w znacznym stopniu złożonością otoczenia, do którego dany program usiłował zaadaptować swoje działanie.

W takim stopniu, w jakim perspektywa ta może być zrealizowana, otwiera się niezwykle ważne pole do symulacji maszynowej jako narzędzia pomocniczego w osiągnięciu głębszego zrozumienia ludzkiego postępowania. Jeżeli bowiem postępowanie to jest określone głównie organizacją części składowych a nie ich fizycznymi właściwościami, zaś maszyny cyfrowe są zorganizowane na podobieństwo człowieka, to w efekcie maszyna może stać się urządzeniem do badania konsekwencji alternatywnych założeń organizacyjnych ludzkiego postępowania”.

I dalej:

„Człowiek, rozpatrywany jako system realizujący pewne postępowanie, jest bardzo prosty. Pozorna złożoność jego postępowania w czasie jest głównie odbiciem złożoności otoczenia, w którym się on znajduje.

...Sądzę, że ta hipoteza jest prawdziwa nawet dla człowieka w ogólności”.

Wiemy już o tym, że tymi aspektami zachowania się maszyn, które nie mogą być przypisane złożoności ich programów, rządzą proste prawa ogólne. W ostateczności są to prawa algebry boole'owskiej. Oczywiście jest też, że własności fizyczne części składowych maszyny są prawie nieistotne dla jej zachowania się. Przekładniki mechaniczne są logicznie równoważone lampom elektronowym, tranzystorom i sztucznym komórkom nerwowym. Oczywiście jest też, że złożoność programów maszyny jest spowodowana złożonością otoczeń branych łącznie z otaczającymi urządzeniami samej maszyny. A programy są tak zbudowane po to, aby sobie z tą złożonością radziły. Jakie mogą być inne jeszcze przyczyny tej złożoności? Tak więc to, co Simon widzi jako perspektywę, jest już realizowane. Ale czy to zebranie oczywistych i prostych faktów prowadzi do wniosku, że człowiek jest tak prosty, jak maszyna? Kiedy Simon dochodzi do tego wniosku, a następnie formułuje zagadnienie tak, jak to zrobił tu, tzn. kiedy sugeruje, że postępowanie „człowieka w ogólności” można rozumieć w kategorii zachowania się maszyn kierowanego przez proste prawa ogólne, wtedy wyklucza on tę jedyną możliwość zrozumienia człowieka, jako istoty autonomicznej, jako indywidualności o głębokich wartościach wewnętrznych. Ale czy można maszynę obrazić? Pytanie „czy mózg jest maszyną z mięsa?”, które stawia Simon w formie tak bardzo wyrafinowanej, jest jednym z rodzaju pytań, które formułuje i może jedynie sformułować mentalność technologiczna. Jeżeli tylko przyjmujemy je za poprawne, rozpoczynają się wściekle spory o to, co maszyna może lub czego „w zasadzie” nie może zrobić. Lecz nie powinna być uznana na samym początku poprawność takiego pytania technologicznego, jak np. czy postępowanie ludzkie ma być rozu-

miane w kategoriach organizacji, czy też własności fizycznych „części składowych”. Zamiast tego można zadać pytanie ludzkie. I rzeczywiście, moglibyśmy zacząć od pytania, co już się stało z „człowiekiem w ogólności”, jeśli może on zaufać maszynom zorganizowanym na swoje podobieństwo. Sukcesy techniki i niektórych wyjaśnień technologicznych oszukują nas już na tyle, że jak sugerowałem, pozwalamy technologii na formułowanie za nas ważnych pytań — pytań, których istota zmniejsza zakres swobody w podejmowaniu przez nas decyzji.

Ktokolwiek dyktuje te pytania, wyznacza jednocześnie w znacznym stopniu i odpowiedzi. W tym sensie technologia, a zwłaszcza technologia maszyn cyfrowych stała się samowystarczającą zmurą, podobną do snu starszej pani, która gwałcona — błaga napastnika, żeby był dla niej uprzejmy. Na to on odpowiada — „to jest twój sen, o pani”. Musimy wreszcie zrozumieć, że technologia jest naszym snem i musimy ostatecznie zdecydować, jak ma się on zakończyć.

Jak to sugerowałem, rewolucja komputerowa nie musi i nie powinna poddawać w wątpliwość ludzkiej godności i autonomii. To pewien rodzaj patologii powoduje, że ludzie aż skracają się od jej nieusprawiedliwionych i bardzo krzywdzących interpretacji. Czy wobec tego komputer jest mniej groźny, niż byśmy sądzili? Jeżeli tylko zdamy sobie sprawę z tego, że nasze wizje, a raczej senne widziadła, wyznaczają wpływ naszych twórców na nas i nasze społeczeństwo, ich groźba dla nas z pewnością zmaleje. Nie można jednak powiedzieć, że sama ta świadomość usunie całe niebezpieczeństwo. Na przykład, poza niszczącym wpływem mentalności technologicznej na wizję człowieka o sobie, istnieją również praktyczne ataki na wolność i godność człowieka, w których decydującą rolę odgrywa technologia komputerów.

Wspomniałem już wcześniej o tym, że nauka o komputerach uznała wagę wprowadzania wiedzy do tych maszyn. Dysponujemy już maszyną DENDRAL [7], która opanowała więcej wiadomości z chemii, niż wielu doktorów chemii oraz inną, o nazwie MATHLAB [8], która opanowała większy zakres matematyki stosowanej, niż wielu matematyków od zastosowań. Zarówno DENDRAL, jak i MATHLAB zawierają wiedzę, którą można oszacować w oparciu o teorie stanowiące podstawę ich konstrukcji. Jeżeli użytkownik sądzi, że rezultat otrzymany z MATHLAB'a jest zły, to wykluczając możliwe błędy programowe, musi on nie zgadzać się nie z maszyną, ani jej programistą, lecz z konkretną teorią matematyczną. Co można jednak powiedzieć o wielu programach, na których opierają się instancje kierujące, w szczególności zaś rząd i woj-

sko, programach, o których w żadnym wypadku nie można twierdzić, że opierają się na uzasadnionych naukowo teoriach. Raczej można powiedzieć, że są one olbrzymim zlepkiem technik programowych, spletych jedynie razem tak, aby działały.

Systemy niezrozumiałe

W naszych usiłowaniach wykorzystania każdego postępu w technice, szybko włączamy do takich zlepków doświadczenia uzyskane z maszynowej obróbki w systemach opartych na teorii. Sądzi się wtedy, że takie zlepki „działają” lepiej. Mam na myśli takie systemy, jak np. systemy wyboru celu używane w Wietnamie, gry wojenne przeprowadzane w Pentagonie itp. Te — często gigantyczne — systemy są składane przez zespoły programistów, pracujące nad tym nie rzadko przez okres wielu lat. Gdy w końcu systemy takie są wdrażane, większość pierwotnych programistów opuściła już swą pracę lub zmieniła zainteresowania. Dość często wtedy, gdy gigantyczne systemy rozpoczynają swe działanie, ich wewnętrzna konstrukcja przestaje być zrozumiała dla każdej jednostki ludzkiej czy małego zespołu ludzi. Norbert Wiener, ojciec cybernetyki, przepowiedział to zjawisko w zadziwiająco przewidującym artykule [9] opublikowanym więcej, niż dziesięć lat temu. Powiedział on tam:

„Może się zdarzyć, że w zasadzie nie potrafimy zbudować żadnej maszyny, której elementy zachowania się nie przestaną być dla nas prędzej czy później niezrozumiałe. Nie oznacza to, w żadnym sensie, że będziemy w stanie zrozumieć te elementy w czasie istotnie krótszym od wymaganego do operowania tą maszyną lub nawet w ciągu dowolnej liczby lat, czy generacji. Inteligentne rozumienie trybu działania (maszyn) może być opóźnione o długi okres czasu w stosunku do zakończenia się zadania ustawionego do wykonywania. Oznacza to, że chociaż maszyny teoretycznie podlegają ludzkiemu krytycyzmowi, to może on być nieefektywny aż do czasu, kiedy przestanie mieć jakiegokolwiek znaczenie”.

Sytuacja, w której się obecnie znajdujemy, ma dwa skutki: po pierwsze, decyzje podejmowane są na podstawie reguł i kryteriów, których nikt bezpośrednio nie zna; po drugie, system reguł i kryteriów staje się nieczuły na zmiany. Dzieje się tak dlatego, że z braku szczegółowego zrozumienia wewnętrznego działania systemu, każda istotna zmiana unieruchomi prawdopodobnie system na dobre. Próg złożoności, poza którym to zjawisko występuje, został już przekroczony przez wiele istniejących systemów, włączając w to niektóre kompilacyjne i operacyjne systemy maszyn cyfrowych. Nikt na przykład nie lubi systemów operacyjnych niektórych wiel-

kich komputerów, lecz nie mogą być one zarówno w sposób istotny zmienione, jak i nie można się bez nich obyć. Zbyt wielu ludzi uzależniło się od nich całkowicie. Niezrozumiały system operacyjny jest niewygodny. Nie to jest jednak najgorsze, ale wzrastająca zależność od supersystemów, które być może zostały zbudowane po to, by pomagać ludziom w dokonywaniu analizy i podejmowaniu decyzji, lecz przekroczyły już próg zrozumienia ich przez użytkowników, stając się jednocześnie narzędziami nie do zastąpienia. A to jest już inna sprawa.

W nowoczesnej wojnie często zdarza się, że żołnierz, np. pilot bombowca, operuje w ogromnej psychologicznej odległości od swych ofiar. Nie jest on odpowiedzialny za spalenie dzieci, ponieważ nigdy nie widzi ich wioski, swoich bomb, a z pewnością i samych płonących dzieci.

Nowoczesna, technologiczna racjonalizacja wojny, dyplomacji, polityki i handlu, jak np. gry z maszyną cyfrową, ma jeszcze bardziej zdradliwy wpływ na posunięcia polityczne. Politycy nie tylko przekazali technologię, której nie rozumieją, swoja odpowiedzialność za podejmowanie decyzji, utrzymując jednocześnie złudzenia, że to oni — twórcy polityki sami formułują zagadnienia polityki i sami je rozwiązują, ale w rzeczywistości odpowiedzialność ta całkowicie się ulotniła. Zaden człowiek nie jest już odpowiedzialny za to, „co maszyna mówi”.

Nie może być wobec tego, zła ani dobra, żadnej kwestii sprawiedliwości, żadnej teorii, z którą się można zgadzać lub nie i — w końcu — żadnej podstawy do krytyki tego „co mówi maszyna”. Mój ojciec zazwyczaj powoływał się na ostateczny autorytet mówiąc do mnie: „tak jest napisane”. Mogłem wtedy jednak przeczytać to, co było napisane, wyobrazić sobie człowieka — autora, wnioskować o jego wartości i — ostatecznie — godzić się lub nie. Systemy w Pentagonie i ich odpowiedniki gdzie indziej w naszej kulturze nie posiadają autorów i to w sensie najbardziej rzeczywistym. Nie dopuszczają więc one do ćwiczeń wyobraźni mogących w końcu prowadzić do ludzkiego osądu. Nic dziwnego, że ludzie, żyjący na codzień z takimi maszynami i popadający w zależność od nich zaczynają wierzyć w to, że ludzie to tylko maszyny. Odbijają oni to, czym stali się sami. Potencjalnie tragiczny wpływ na społeczeństwo, który może wynikać ze stosowania omówionych przeze mnie systemów, jest znacznie większy, niż to można sobie od razu wyobrazić. I znów liczą się najbardziej nie efekty bezpośrednie, lecz skutki uboczne.

Po pierwsze, istnieje oczywiście psychologiczny wpływ na jednostki żyjące w społeczeństwie, w którym pewne siły anonimowe, a więc nieodpowiedzialne, formułują więk-

szość pytań codziennych i określają zakres możliwych na nie odpowiedzi. Nie można się dziwić, że duża liczba jednostek delikatnych, żyjących w takim społeczeństwie, doświadcza pewnego rodzaju impotencji i pada ofiarą bezrozumnych ataków.

Co gorsza, z tego względu, że oparte na maszynach cyfrowych systemy wiedzy stają się w zasadzie nieczułe na zmiany, poza tym, że mogą one rosnać, a także dlatego, że wprowadzają one stan zależności i nie można ich, po przekroczeniu pewnego progu, odrzucić — istnieje duże ryzyko, że przekazane zostaną one następnym generacjom, w zwiększającej się stale objętości.

Ludzie również przekazują wiedzę następnym generacjom. Ponieważ jednak człowiek jest śmiertelnym, więc jego przekazywanie wiedzy następnym generacjom jest od razu procesem filtracji i przenoszenia. Człowiek nie przekazuje jedynie wiedzy, on ją raczej ciągle generuje. Pomimo żalu nad upadłymi cywilizacjami starożytnymi wiemy, że chwała człowieka leży zarówno w ewolucji jego kultur, jak i rozwoju jego umysłu. Nierozsądne stosowanie coraz to większych i bardziej złożonych systemów obliczeniowych może doprowadzić do zahamowania tego procesu. Może spowodować zastąpienie przyływów i odpływów kultury przez świat bez wartości, świat faktów, które dawno temu zostały określone i ustalone na zawsze.

Efekty pozytywne

Mówiłem uprzednio o pewnych potencjalnie niebezpiecznych aspektach obecnych kierunków rozwoju systemów obliczeniowych. Czy na ten temat nie można powiedzieć nic pozytywnego? Tak, lecz należy formułować to ostrożnie. Tu ponownie skutki uboczne są ważniejsze od efektów bezpośrednich. Szczególnie idea obliczeń i języków programowania staje się ważną metaforą, która w przyszłości może okazać się odpowiedzialną za zmiany w sposobie interpretacji w wielu różnych dziedzinach.

Większość najbardziej podstawowych wzorów myślowych, za pomocą których ludzie interpretują zjawiska, zarówno fizyczne jak i społeczne, tkwi ciągle w podstawowych analogiach mechanicznych. Dla przykładu, układy dynamiczne Marksa, jak również i Freuda, są w zasadzie systemami zrównoważonymi. Każdy hydrodynamik jest w stanie zrozumieć je bez potrzeby wychodzenia poza granice swego żargonu zawodowego.

Języki zdolne do opisywania procesów czynnych, a w szczególności do opisanie ich w postaci modularnych podprocesów, posiadają już ogromny wpływ na sposób myślenia ludzi zajmujących się maszy-

nami cyfrowymi, mówią o każdym aspekcie ich światów, nie tylko związanych z ich pracą. Tak zrodzony pogląd na świat przetwarzania informacji wyraża pewną swoją metaforę. Świadczy o tym fakt, że: 1) tworzy on pewien system myślowy, który pozwala na zadawanie nowych pytań dotyczących szerokiego wachlarza zjawisk oraz 2) pogląd ten sam dostarcza kryteriów wystarczalności otrzymywanych odpowiedzi.

Nowa metafora jest ważna nie dlatego, że może ona rozszerzyć spojrzenie człowieka przez zaferowanie mu dodatkowego punktu widzenia swego świata. Efektywność nowego podejścia może rzeczywiście skusić leniwe umysły do przyjęcia go za podstawę uniwersalnych wyjaśnień i źródło uniwersalnych recept.

Proces maszynowej symulacji zjawisk społecznych został już tak rozwinięty przez niezdolnych do szerszego spojrzenia badaczy, że według nich pozwala ona na otrzymywanie ogólnych rozwiązań wszystkich problemów ludzkości.

Przekazane nam przez religię, poetów i takich filozofów, jak Darwin, Newton, Freud i Einstein metafory dość szybko przeniknęły do języka zwykłych ludzi. Metafory te stały się więc narzędziem kształtowania przez naszą cywilizację wyobrażenia o świecie. Metafora przetwarzania informacji dostępna jest na razie tylko wąskiemu kręgowi społeczeństwa.

Przyswojenie jej i wcielenie, jako tylko jednego z wielu poglądów na świat, wymaga pewnego rodzaju doświadczenia i obycia z systemami obliczeniowymi. Być może, że takie obycie rozpowszechni się bardzo w zaawansowanych technicznie kręgach społeczeństwa krajów najbardziej rozwiniętych. Lecz jeśli stanie się ona dominującym trybem myślenia pewnych tylko klas społecznych, to okaże się wtedy nie tylko represyjną w zwykłym tego słowa znaczeniu, lecz również i siłą społeczną ogromnie dzielącą ludzi. Stanie się tak dlatego, że zarówno klasy, dla których metafora ta będzie dostępna, jak i te, dla których będzie ona niedostępna, stracą w bardzo istotnym sensie możliwość wzajemnego porozumienia się. Powszechnie jest już wiadomo, jak trudno jest porozumiewać się biednym i wyszkiwanym z resztą społeczeństwa, którego są przecież częścią.

Wiemy też, jak trudno porozumiewać się ludziom z kręgu nauk ścisłych i technicznych z ludźmi z kręgu sztuki i nauk humanistycznych. W obu tych przypadkach trudności porozumienia się, pociągające często za sobą bardzo przykre skutki, spowodowane są głównie faktem posiadania przez dane grupy społeczne rozłącznych doświadczeń, z których wyrosły rozłączne metafory.

Odpowiedzialność

Zdając sobie sprawę z tych ograniczonych możliwości, zastanówmy się, jaka powinna być odpowiedzialność informatyków?

Przed wszystkim muszę powiedzieć, że znaczna część potencjalnego złego wpływu maszyn cyfrowych jest bardziej funkcją właściwości przypisywanych maszynom przez ludzi, aniżeli aktualnych możliwości tych maszyn wynikających z ich konstrukcji.

Ludzie innych zawodów mogą tworzyć swe wyobrażenia o maszynach cyfrowych jedynie w oparciu o propagandę głoszoną przez naukowców informatyków, odpowiednio wzmocnioną przez prasę. Niezwykle ważną odpowiedzialnością naukowca informatyka jest więc umiarkowane wygłaszanie swych poglądów.

Rady takiej nie trzeba byłoby nawet formułować, gdyby nauka o maszynach cyfrowych posiadała tradycję takiego poziomu naukowego i samokrytycyzmu, jaka cechuje starsze dziedziny nauki. Dojrzały naukowiec jest pełen szacunku dla głębi swego przedmiotu badań. Ta właśnie skromność jest źródłem jego własnej siły. Uważam, że wpajanie stylu takiej skromności, głównie przykładem osobistym, jest jedną z najważniejszych misji każdego uniwersyteckiego wydziału nauki o maszynach cyfrowych.

Naukowiec informatyk musi stale zdawać sobie sprawę z tego, że jego narzędzia wywierają ogromne skutki, tak bezpośrednie, jak i pośrednie. Na przykład, błąd w programie może spowodować żałosne skutki bezpośrednie, włącznie ze znacznymi ofiarami w ludziach. Dla przykładu podam, że 11 września 1971 roku błąd w programie maszyny cyfrowej spowodował równoczesne zniszczenie 117 wysokościowych balonów meteorologicznych, których instrumenty były kontrolowane przez satelitę Ziemi [10].

Podobny błąd w programie systemów dowodzenia i kontroli wojсковой mógłby spowodować wysłanie całej floty rakiet z głowicami nuklearnymi. Jedynie cenzura prasowa nie pozwala nam dowiedzieć się, ile wydarzyło się już podobnych wypadków z bronią konwencjonalną. Jest więc sprawą oczywistą, że naukowiec informatyk jest istotnie odpowiedzialny za cał-

kowite wyjaśnienie kwestii zawodności i ograniczeń systemów, które jest on w stanie zaprojektować. Istotna siła jego systemów powinna polegać na zdolności powstrzymywania się od rad, które gotów jest udzielić oraz ograniczania zakresu prac, które chce on wykonać. Naukowiec informatyk, tak jak i wszyscy ludzie, jest oczywiście odpowiedzialny za swoje czynności i ich konsekwencje. Czasami trudno jest przyjąć na siebie taką odpowiedzialność, ponieważ o tym, co należy, a czego nie należy robić wydaje się decydować jakiś autorytet siły odległej i anonimowej. Tym smem, z którego należy się obudzić, jest wiara w technologię, która sama określa to, co ma być zrobione poprzez procesy ekstrapolacyjne, oraz w to, że jednostka jest bezsilna w próbach przeciwstawiania się temu samookreśleniu. Rozważmy ogromne systemy maszyn cyfrowych. Stanowią one oczywiście naturalną ekstrapolację dużych systemów już istniejących. Sieci powiązanych ze sobą systemów obliczeniowych stanowią dalszy krok tej ekstrapolacji. Można się zastanawiać, czy takie systemy mogą być używane jedynie przez organizacje państwowe i bardzo duże zrzeszenia przedsiębiorstw, a także czy przypadkiem takie organizacje nie będą używały tych systemów głównie do działalności skierowanej przeciwko ludziom.

Rozważmy też systemy rozpoznawania mowy. Czyż nie będą one używane głównie do szpiegowania prywatnych połączeń? Odpowiedź na te pytania, uznającą duże systemy obliczeniowe, sieci systemów obliczeniowych oraz systemy rozpoznawania mowy za rzecz konieczną, należy uznać za wyraz podporządkowania czyjejs godności ludzkiej. Odpowiedź taka musi bowiem opierać się na czymś głębokim przekonaniu, że społeczeństwo straciło już władzę nad swoją technologią albo też pogląd taki wynikać może z całkowicie niemoralnej postawy, że „jeśli ja tego nie zrobię, to zrobi to ktoś inny”. Nie twierdzę wcale, że wspomniane przeze mnie systemy muszą być konieczne złe. Sądzę tylko, że mogą być one złe, a co najważniejsze,

że konieczność ich istnienia nie może być uznana przez jednostki głoszące swoją autonomię, wolność i godność. Każdy naukowiec informatyk może i powinien decydować. Przynajmniej częściowo od niego zależy określenie zakresu wpływu maszyn cyfrowych na społeczeństwo.

Podstawowym w końcu pytaniem, które musi sobie zadać naukowiec informatyk, jest pytanie, które musi zadać sobie każdy naukowiec, a właściwie każdy człowiek. Nie jest to pytanie „czego powinienem dokonać?”, lecz raczej — „kim powinienem zostać?”. Na pytanie to mogę odpowiedzieć tylko za siebie.

Ponownie jednak chcę podkreślić, że jeśli technologia jest złym smem zdającym się posiadać własną logikę konieczności, to jest to nasz zły sen. Człowiek obdarzony logiką i siłą wewnętrzną jest w stanie odebrać technologii prawo formułowania zagadnień ludzkich. Człowiek może sam formułować ludzkie zagadnienia i otrzymywać ludzkie rozwiązania.

Bibliografia

- [1] B. F. SKINNER: Beyond Freedom and Dignity. Knopf, New York, 1971.
- [2] K. M. COLBY, S. WEBER, F. D. HILF: Artificial Intelligence. 1, 1/1971.
- [3] N. CHOMSKY: Aspects of the Theory of Syntax. M.I.T. Press, Cambridge, Mass., 1965.
- [4] M. HALLE: The Sound Pattern of English. Harper Row, New York, 1968.
- [5] L. MUMFORD: The Pentagon of Power. Harcourt. Brace. Jovanovich, New York, 1970.
- [6] H. A. SIMON: The Sciences of the Artificial. M.I.T. Press, Cambridge, Mass., 1969, str. 22—25.
- [7] B. BUCHANAN, G. SUTHERLAND, E. A. FEIGENBAUM: Machine Intelligence. B. Meltzer, Ed. American Elsevier, New York, 1969.
- [8] W. A. MARTIN, R. J. FATEMAN: "The Macsyma system" (Association for Computer Machines, New York, 1971), J. Moses Communications of the A.C.M. 14, Nr 8/548 (1971).
- [9] N. WIENER, Science 131, 1355 (1960).
- [10] R. GILLETTE, Science 174, 477 (1971).

Tłum. i oprac. Ryszard Sienkiewicz na podstawie artykułu J. Weizenbauma pt. „On the impact of the computer on the society”, SCIENCE nr 4035 z 1972 r.

KALENDARZ IMPREZ ZAGRANICZNYCH

Data	Impreza	Miejsce	Organizator — Informacje
19—21. 07.74	VII Journées Internationales de l'Informatique et de l'Automatisme	Paryż Francja	Commissaire Général des JIIA 6, rue Dufrenoy, 751 16 Paris, France
21—20.07.74	International Congress of Mathematicians	Vancouver Kanada	Prof. R. Jamer, Univ. of British Columbia, Vancouver 8, Canada
23—26.07.74	European Conference on Circuit Theory and Design	Londyn W. Brytania	The Institution of Electrical, Savoy Place, London Wc2 R OBL, England
29.07—2.08.74	Second Jerusalem Conference on Information Technology	Jerozolima Izrael	ILTAM Limited, POB 7170, Jerusalem, Israel

Programista jako navigator

Amerykańskie Stowarzyszenie Association for Computing Machinery (ACM), będące niewątpliwie największą na świecie organizacją grupującą informatyków, na każdy ze swoich dorocznych kongresów zaprasza wybitnego specjalistę w danej dziedzinie z prośbą o wygłoszenie wykładu na wybrany temat. Są to tzw. Turing Award Lectures. Ubiegłoroczny wykład wygłoszony został przez Charlesa W. Bachmana, jednego z pionierów nowoczesnej technologii baz danych. Był on twórcą jednego z trzech najczęściej używanych dzisiaj systemów zarządzania bazami danych (IDS), a następnie jednym z założycieli grupy CODASYL. Obecnie pracuje nad metodami opisywania i ujawniania relacji w zbiorach danych.

W 1973 roku cały świat obchodził pięćsetną rocznicę urodzin Mikołaja Kopernika, wielkiego polskiego astronoma i matematyka. Jego heliocentryczny model naszego układu planetarnego stał w jaskrawej sprzeczności ze zbudowaną 1400 lat wcześniej geocentryczną teorią Ptolemeusza. Teoria Kopernika poddana została miazdzącej i natarczywej krytyce, a jeszcze 10 lat później Galileusz został zmuszony przez inkwizytorów do oświadczenia, że przestał wierzyć w model kopernikański.

Przytaczam te dobrze znane fakty dla podkreślenia podobieństwa sytuacji, która — moim zdaniem — istnieje dzisiaj w świecie ptolemejskich systemów informacyjnych. Systemy te oparte były na „komputerocentrycznym” sposobie myślenia. Tak, jak starożytni widzieli Ziemię i krążące wokół niej Słońce, tak „starożytni” naszych systemów informacyjnych spoglądali głównie komputer z przepływającym przez niego, sekwencyjnym zbiorem danych. Każdy z tych modeli był adekwatny do swojego czasu i miejsca; każdy został później skrytykowany, jako nieodpowiedni i zastąpiony innym modelem, który dokładniej odzwierciedlał rzeczywisty świat i jego zachowanie.

Punkt widzenia Kopernika dał podstawy nowoczesnej mechanice niebios i pozwolił na zrozumienie tajemniczych trajektorii planet.

Nowy sposób rozumowania w dziedzinie systemów informacyjnych przyjmuje za centrum nie komputer, a bazę danych. To nowe podejście pozwoliło na otrzymanie nowych rozwiązań naszych problemów dotyczących baz danych i przyspieszyło podbój n-wymiarowej przestrzeni struktur danych, lepiej modelującej złożoność świata realnego.

W technologii zbioru sekwencyjnego technika wyszukiwania polegała na określeniu pierwotnego klucza, a następnie na przeglądaniu po kolei zapisów (rekordów) składających się na zbiór, aż do odnalezienia zapisu o poszukiwanej wartości klucza. Kluczem pierwotnym nazywamy takie pole zapisu, które wyznacza go jednocześnie w zbiorze. Przykładami takich kluczy mogą być wszelkie symbole identyfikacyjne,

numery kart i transakcji itp. Prawie bez wyjątku są to wielkości sztuczne, lecz zbudowane tak, aby mogły być jednoznacznie interpretowane. Atrybuty naturalne (nazwiska i nazwy, daty i ilości) nie były używane, ponieważ nie zapewniały jednoznaczności.

Wprowadzanie pamięci pomocniczych o dostępie bezpośrednim wywołało w sposobie widzenia rzeczy przez programistę zmianę porównywalną do kopernikańskiej. Przestał on być statycznym obserwatorem przepływu obiektów w pamięci operacyjnej, a stał się ruchliwym navigatorem zdolnym do przemierzania i zagłębiania się w bazę danych.

Nowe rodzaje pamięci umożliwiły przejście do innych metod wyszukiwania zapisu według klucza pierwotnego. Pierwsza z tych metod, zwana randomizacją (has technique), polega na wylizaniu — za pomocą specjalnego algorytmu zastosowanego do klucza — adresu w pamięci, pod którym spo-

dziewamy się znaleźć poszukiwany zapis.

Pewną alternatywą do metody randomizacji jest indeksowo-sekwencyjna metoda dostępu, która również używa klucza pierwotnego oraz wieloszczęblowych indeksów (stosujemy ją na co dzień przy poszukiwaniu hasła w wielotomowej encyklopedii).

Nowe metody pozwoliły na daleko idącą redukcję średniego czasu dostępu do zapisu. Jednakże programista, mając do dyspozycji tylko jeden klucz pierwotny, porusza się ciągle w przestrzeni jednowymiarowej.

Wychodząc z tego stanu rzeczy, chciałbym rozpocząć trening programisty — doświadczonego nawigatora — w n-wymiarowej przestrzeni danych. Zanim jednak opiszę ten proces, chciałbym powiedzieć na czym, moim zdaniem, polega zarządzanie bazą danych.

Zajmuje się więc ono wszelkimi aspektami zapamiętywania, wyszukiwania, modyfikacji i wykreślenia danych ze zbiorów używanych wielokrotnie i ak-



tualizowanych wraz z przybywaniem nowych informacji. Zarządzanie bazą danych ma dwie podstawowe funkcje. Pierwsza z nich polega na wyszukiwaniu uprzednio zapamiętanych danych w celu stwierdzenia stanu obiektu należącego do świata rzeczywistego, opisywanego przez te dane. Do funkcji zarządzania należy tu także przygotowywanie wyników czytelnych dla człowieka.

Drugą funkcją zarządzania bazą danych jest jego aktualizacja. Obejmuje ona początkowe zapamiętywanie danych, ich modyfikację w miarę zmiany stanu rzeczy i wreszcie — usuwanie danych ze zbioru wtedy, gdy nie są potrzebne. Inną czynnością, która pochłania wiele czasu komputera jest sortowanie zbiorów.

Powróćmy teraz do naszego programisty — nawigatora.

Gdy opuściliśmy go, używał właśnie metody randomizacji (lub metody indeksowo-sekwencyjnej), aby na podstawie wartości klucza pierwotnego uzyskać dostęp dożądanego zapisu w zbiorze.

Jednakże często potrzebna jest możliwość wyszukiwania zapisów w zbiorze na podstawie wartości innych jego pól.

Chcemy więc np. wyszukać zapisy dotyczące tych pracowników, którzy przystąpili do pracy w przedsiębiorstwie w 1964 r. Wyszukiwanie odbywać się więc będzie za pomocą klucza wtórnego. Ilość zapisów w zbiorze o tej samej wartości klucza wtórnego może wynosić od zera do liczby elementów zbioru. Wraz z wprowadzeniem klucza wtórnego przestrzeń danych — poprzednio jednowymiarowa — zyskała nowy wymiar, równy liczbie pól w zapisie. Różnica pomiędzy bazą danych, a zbiorem danych nie jest ściśle określona, jednakże jedna cecha odróżniająca zasługuje w tym miejscu na podkreślenie. Mianowicie w bazie danych mamy wiele rodzajów zapisów.

I tak np. w kartotece personelu przedsiębiorstwa mamy zapisy opisujące pracowników, wydziały, kwalifikacje, wykształcenie itp. Każdy typ zapisu ma swój jednoznaczny klucz pierwotny, a każde z pozostałych pól jest potencjalnym kluczem wtórnym. Klucze pierwotne i wtórne wchodzą tu w interesujące relacje. Sytuacja, w której klucz pierwotny jednego typu zapisu jest kluczem wtórnym innego typu zapisu odzwierciedla relacje, które zachodzą między obiektami świata rzeczywistego.

Ta równoważność kluczy była podstawą do definiowania zbiorów struktur danych, na tej koncepcji oparty został system Integrated Data Store (IDS). W systemie tym programista miał możliwość łączenia zapisów w zbiory struktur danych, a następnie użycia definicji tych zbiorów jako „ścieżki wyszukiwanej” interesujące go zapisy.

Wyobraźmy sobie, jak przebiega konstrukcja takiej ścieżki wyszukiwanej. Każde wejście do bazy danych zawiera pewien klucz pierwotny, który określi zapis stanowiący punkt wejścia do bazy danych. Zapis ten będzie wykorzystany dla dojścia do innych zapisów zbioru, z których każdy posłuży, jako punkt wyjścia do zbadania innego zbioru.

Przypuśćmy, że w oparciu o kartotekę personelu przedsiębiorstwa należy sporządzić listę pracowników wybranego wydziału. Kartoteka zawiera dwa typy zapisów: dla personelu i dla wydziału. Pierwszy z nich zawiera trzy pola: numer pracownika (klucz pierwotny dla zapisu), jego nazwisko i numer wydziału; drugi ma tylko dwa pola: numer wydziału (klucz pierwotny) i nazwę wydziału. Wspólne użycie numeru wydziału przez oba typy zapisów oraz zadeklarowanie zbioru opartego na tym kluczu umożliwia stworzenie relacji pomiędzy zapisem wydziału, a zapisami przedstawiającymi pracowników wydziału.

Dodanie w zapisie wydziału numeru jego kierownika zwiększa możliwości nawigacji i pozwala na zdefiniowanie drugiej klasy zbiorów. Mając dany numer pracownika, można zażądać listy kierowanych przezeń wydziałów, następnie ich pracowników; kierowanych przez nich wydziałów itd.

Podobnie, na podstawie numeru pracownika możemy zidentyfikować jego szefa itd. aż do dyrektora przedsiębiorstwa.

Przed nawigatorem, poruszającym się w bazie danych, stoją też innego typu problemy, wynikające z faktu wspólnego korzystania z bazy przez wielu użytkowników. Dostęp dzielony jest nową, złożoną formą wieloprogramowania. W tradycyjnym systemie wieloprogramowym autor programu nie musiał się troszczyć o to, że wraz z innymi dzieli pamięć, procesor i pozostałe składniki komputera — o to wszystko dbał system operacyjny.

W bazach danych do listy dzielonych zasobów trzeba jeszcze dopisać dane, dzielone pomiędzy wielu użytkowników. Stwarza to cały szereg problemów — wystarczy wyobrazić sobie np. dwa programy, usiłujące zająć ostatnie wolne miejsce na dany lot w systemie rezerwacji miejsc lotniczych.

Istnieją dwa zasadnicze powody, dla których problemy dostępu dzielonego zasługują na szczególną uwagę. Pierwszym jest trend do scalania wielu zbiorów danych w bazy danych; w tej sytuacji cała baza danych byłaby zablokowana dopóty, dopóki pojedynczy, pracujący w systemie partiovym program (lub — co gorsza — pracujący interakcyjnie programista) nie zakończyłby pracy. Drugim powodem jest względnie długi czas dostępu do pamięci pomocniczej; jest to szczególnie odczuwalne w sytuacji, w której odwoływanie się do

pamięci następuje w wyniku realizacji cyklu: odwołanie, proces, odwołanie, proces... Naturalnym rozwiązaniem jest tutaj jednoczesna obsługa wielu odwołań do pamięci pomocniczej na raz, co właśnie prowadzi do dostępu dzielonego.

Z dwóch wymienionych uprzednio, podstawowych funkcji zarządzania bankiem danych jedynie aktualizacja może stwarzać potencjalne problemy dostępu dzielonego. Nieograniczona liczba programów może odczytywać zawartości zapisów. Jednakże w chwili, gdy jeden program zaczyna aktualizować wybrany zapis, mogą rozpocząć się kłopoty.

Nadejdzie bowiem moment, w którym dwa programy będą jednocześnie chciały przystąpić do opracowywania tego samego zapisu.

Dwa podstawowe kłopoty, które możemy napotkać przy dostępie dzielonym, nazwiemy interferencją i kontaminacją. Z interferencją mamy np. do czynienia wtedy, gdy jeden program dokonuje bilansu konta bankowego w danej chwili, a drugi równocześnie dokonuje nowych obciążeń tego konta.

W sytuacji interferencji program, który ją spowodował, musi być najpierw zatrzymany, a następnie ponownie zainicjowany, aby dać mu nową możliwość doprowadzenia do wyniku. Wszelkie rezultaty, otrzymane przed tym wynikiem muszą być usunięte, ponieważ otrzymane będą nowe wyniki.

Z kontaminacją spotykamy się wtedy, gdy program z jakichś przyczyn (np. interferencji) został usunięty z systemu, lecz jego wynik został wczytany przez inny program. W tej sytuacji nie wystarczy ponowne inicjowanie pracy pierwszego programu; niezbędne jest usunięcie i ponowne inicjowanie drugiego programu, tak, aby mógł on operować poprawnymi danymi wyjściowymi.

Krytycznym problemem, pojawiającym się przy rozwiązywaniu zagadnień dostępu dzielonego, jest: do jakiego stopnia programista powinien mieć wgląd w metodę dostępu? Najstuszniejszym założeniem jest przyjęcie zasady, że programista nie powinien w ogóle być świadomy problemów, jakie stwarza dostęp dzielony. W ten właśnie sposób zbudowany został system firmy Weyerhaeuser.

System ten automatycznie blokuje aktualizowany zapis oraz wszelkie sygnały wysyłane przez program aktualizujący, aż do jego normalnego zakończenia.

Rozwiązanie to kompletnie eliminuje problem kontaminacji. Ma ono jednak ten skutek uboczny, iż może spowodować sytuację „martwej ciszy” (*deadly embrace*) w działalności systemu. Dzieje się tak wtedy, gdy dwa lub więcej programów czeka na siebie wzajemnie, aby odblokować dany zapis. W systemie IDS przerywa się program, który spowodował całą tę sytuację, a następnie ponownie zapamiętuje się zapisy zmie-

nione przez ten program, po czym u-
dostępna się je oczekującym progra-
mom. Wreszcie na końcu ponownie
inicjuje się działanie przerwane-
go programu.

Czy tego typu sytuacje zdarzają się
często? We wspomnianym systemie
Weyerhaeuser około 10 procent progra-
mów musiało być ponownie inicjowa-
nych z powodu sytuacji „marwej ci-
szy” (oznacza to około 100 programów
na godzinę).

Można postawić następujące pytania:
— czy uniknięcie dostępu dzielonego
pozwoliłoby na wykonanie mniejszej
(większej) liczby programów, w ciągu
godziny?

— czy inna strategia (np. raczej wy-
krywanie kontaminacji, niż jej usuwa-
nie) nie byłaby skuteczniejsza?

— czy uświadomienie programiście pro-
blemów dostępu dzielonego pozwoliłoby
na ominięcie problemu i podniesienie
efektywności?

Chciałbym tu zaproponować, aby dzi-
siejsi programiści odeszli od „pamięcio-
centrycznego” sposobu widzenia świata
i przeszli do nawigacji w n-wymiaro-
wej przestrzeni danych. Niezbędne po
temu narzędzia programistów już ist-
nieją i są coraz szerzej dostępne.

Bertrand Russel, wybitny filozof i
matematyk angielski zauważył, że
sformułowanie teorii względności
spowodowało konieczność zmiany
naszych wyobrażeń o świecie.

Podobne zmiany pożądane są w
naszym wyobrażeniu o świecie sy-
stemów informatycznych.

Największym problemem jest reor-
ientacja myślenia ludzi zajmujących
się przetwarzaniem danych.

Chodzi tu nie tylko o programis-
tów, ale o projektantów systemów,
planistów oraz programistów syste-
mowych tworzących systemy ope-
racyjne i inne elementy oprogra-
mowania baz danych.

Koszt wyposażenia sprzętowego i
programowego baz danych, które
zostaną zrealizowane w latach
osiemdziesiątych, szacuje się obec-
nie na 100 miliardów dolarów.

Brak skutecznej standaryzacji spo-
woduje wzrost tej sumy o dalsze
20 miliardów. Rozsądnie było by
więc odrzucić konserwatyzm, emo-
cjonalizm i argumenty teologiczne,

które w obecnej chwili opóźniają
postęp.

Uczelnie nie miały dotychczas nic
do powiedzenia na temat wielkich
baz danych; tego rodzaju przedsię-
wzięcia nie leżą w zasięgu ich
możliwości finansowych. Niezbędne
będą w tej dziedzinie wspólne u-
czelniano-rządowe projekty badaw-
cze, które pozwolą stworzyć nie-
zbędne podstawy postępu. Nie mam
tu na myśli np. badań nad nowymi
algoritmami randomizacji; chodzi o
stworzenie mechanizmów dla niemal
miliarda znaków z realnego życia
gospodarczego, zorganizowanych w
najczystsza ze struktur danych, ja-
kie dziś znamy.

Rozpoczął się powszechny dialog,
mam nadzieję, że będzie on konty-
nuowany. Jeżeli patronować mu
będzie atmosfera, w której żadna
organizacja nie będzie usiłowała
dominować nad rozważaniami, to
pewien jestem, że wyposażymy
programistę w skuteczne narzędzia
nawigacyjne.

Tłum. i oprac. Stanisław Matwin

PRZEGLĄD WYDAWNICTW

Bibliografia wydawnictw polskich z dziedziny informatyki

● Architektura poliprzetwarzania — Świt czwartej ge-
neracji. Tłum. wyd. ang. z roku 1972. Wyd. Ośrodka Badaw-
czo-Rozwojowego Informatyki INFORNA, Warszawa 1973,
ss. 69, Europejski Program Badawczy Diebolda. Zeszyt 48
(E-100), cena zł 92.—

Dotychczasowy rozwój i ogólne trendy rozwojowe systemu.
Architektura poliprocessora: właściwości, hardware, soft-
ware, łącza. Architektura przetwarzania i projektowania,
typowy system poliprocessora, wpływ na zarządzanie ośrod-
kiem przetwarzania danych, alternatywy.

Materiały przeznaczone są dla projektantów systemów EPD.

● Przeglądy kontrolne systemów. Tłum. wyd. ang. z roku
1971. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki IN-
FORNA, Warszawa 1973, ss. 93. Europejski Program Badaw-
czy Diebolda. Zeszyt 40 (E-86), cena zł 92.—

Administracyjne środki kontroli: organizacyjne środki kon-
trolli, procedury wspierające, zapobieganie katastrofie i wy-
prowadzanie systemu z katastrofy. Środki kontroli syste-
mu: środki kontroli wejścia, zaprogramowane środki kon-
trolli, środki kontroli przetwarzania w systemach pracują-
cych na bieżąco, środki kontroli wyjścia, dokumentacja.
Zabezpieczenie: zabezpieczenie fizyczne, eksploatacja itd.
Inspektor wewnętrzny WAPD: udział inspektora, inspektor
APD.

Załączniki: obowiązki inspektora wewnętrznego, ochrona
przeciwpożarowa systemów komputerowych, lista kontroli
na inspekcji systemu.

Materiały przeznaczone są dla inspektora wewnętrznego
APD.

● Wytyczne do planowania działalności APD. Tłum. Wyd.
ang. z roku 1972. Wyd. Ośrodka Badawczo-Rozwojowego In-
formatyki INFORNA, Warszawa 1973, ss. 112. Europejski Pro-
gram Badawczy Diebolda. Zeszyt 52 (E-96), cena zł 92.—

Wprowadzenie: organizacyjne przygotowanie planowania
działalności APD, metodologia itd. Wytyczne do optymaliz-
acji efektywności planowania działalności informatycznej:
charakterystyka kluczowych działań gospodarczych przed-
siębiorstwa, uogólnienie informacji o kluczowych dzia-
łaniach gospodarczych, określenie planu rozwoju systemów
informatycznych, formułowanie strategii systemów infor-
matycznych, Ramowy Harmonogram Realizacji Systemów,
nadzorowanie wykonania planu. Czynniki decydujące o
prowadzeniu planowania: minimalizacja ryzyka związanego
z planowaniem APD, cechy dobrego planu APD, uzyski-
wanie aprobaty ze strony naczelnego kierownictwa przed-
siębiorstwa.

Materiały przeznaczone są dla projektantów (planistów)
systemów EPD i dla personelu kierowniczego przedsię-
biorstw.

● Kryteria oceny działalności ośrodka APD. Tłum. wyd.
ang. z roku 1971. Wyd. Ośrodka Badawczo-Rozwojowego In-
formatyki INFORNA, Warszawa 1973, ss. 71. Europejski Pro-
gram Badawczy Diebolda. Zeszyt 39 (E-87), cena zł 92.—

Wydatność produkcyjna ośrodka APD. Czynniki, które
wpływają na wydajność, podejście do zagadnienia oceny
uzyskanej wydajności oraz zdolności produkcyjnej, pomoce
w zakresie oceny, zalecenia dla kierownictwa działów eks-
ploatacji w zakresie ustalenia i zwiększenia wydajności.
Streszczenie wyników przeglądu. Podejście do zagadnienia
oszacowania i optymalizacji parametrów eksploatacyjnych.
Materiały przeznaczone są dla personelu kierowniczego
ośrodków obliczeniowych.

● Maszyny analogowe budowa i zastosowanie — MALZA-
CHER S. Wyd. 2. popr. i uzupełn. Wyd. Uniwersytetu Śląskie-
go, Warszawa 1973, ss. 72, cena zł 8.—

Zasada działania m.a. Człony operacyjne m.a. Generatory
funkcji wymuszających. Konstrukcja i wyposażenie m.a.
Zastosowanie m.a. Dodatek (parametry scalonych wzmac-
niaczy operacyjnych).

Skrypt przeznaczony jest dla studentów Wydziałów Tech-
niki Uniwersytetu Śląskiego oraz dla słuchaczy Wyższych
Szkół Pedagogicznych.

INFORMATYKA z wizytą w MERAMACIE

Redakcja **INFORMATYKI** została zaproszona do zwiedzenia intensywnie rozbudowujących się zakładów przemysłu sprzętu informatyki i ich zaplecza naukowo-badawczego. Wizyty te mają na celu bezpośrednie zaznajomienie redakcji z dotychczasowym dorobkiem i zamierzeniami tych zakładów, wzmocnienie kontaktów z producentami i ustalenie ciągłych form rzetelnego informowania Czytelników naszego czasopisma o obecnych możliwościach przemysłu oraz o realizowanym programie rozwoju zaopatrywania użytkowników w sprzęt produkcji krajowej.

Ostatnio przedstawiciele redakcji odwiedzili Warszawskie Zakłady Urządzeń Informatyki **MERAMAT**. Zwiedzaniem tego zakładu — jednego z najmłodszych w branży — pokierował osobiście dr inż. Zdzisław Łapiński — organizator i dawny dyrektor **MERAMATU**, a obecnie I Zastępca Dyrektora Naczelnego Zjednoczenia **MERA**. Dyrektor Zakładu inż. Stanisław Górczyński wraz z I zastępcą, inż. Zbigniewem Białczykiem, jak również kierownicy wydziałów produkcyjnych i rozwojowych, poświęcili dużo czasu przedstawicielom naszej redakcji, aby pokazać produkcję i prowadzone prace rozwojowe. Opowiedzieli oni przy tym wiele interesujących szczegółów o osobistym

wkładzie dr inż. Łapińskiego w uruchamianie od podstaw Zakładu od 1970 r. i w konsekwentną jego rozbudowę.

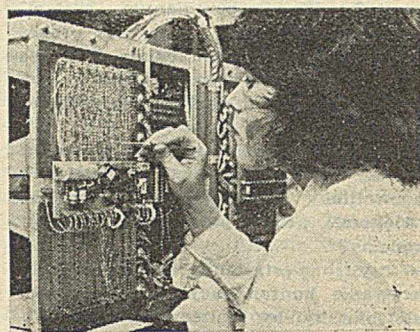
Obecnie Zakład robi — nawet przy powierzchownym zwiedzaniu — duże wrażenie, przede wszystkim ze względu na panujący tam porządek i widoczne już metody nowoczesnej organizacji pracy. Krótka ta przecież historia **MERAMATU** zasługuje na dokładniejsze przeanalizowanie w celu upowszechnienia doświadczeń.

Były też innego rodzaju wrażenia. Redaktor naczelny **INFORMATYKI** prof. dr hab. Leon Łukaszewicz mógł naocznie stwierdzić, jak zainicjowane przez niego jeszcze w końcu lat 50-tych w Zakładzie Aparatów Matematycznych (później Instytut Maszyn Matematycznych) — prace badawcze w dziedzinie pamięci taśmowych i głowic magnetycznych — obecnie owocują dzięki wprowadzeniu tych wyrobów do produkcji w **MERAMACIE** i niestannemu ich rozwojowi.

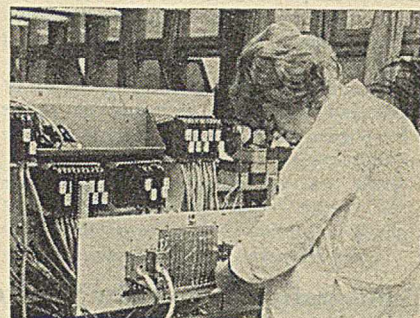
Dziś **MERAMAT** jest jedynym w kraju producentem pamięci taśmowych oraz głowic magnetycznych do pamięci taśmowych i bębnowych.

Pamięci taśmowe **PT-3** znajdują powszechne zastosowanie jako pamięci zewnętrzne komputerów

Rys. 1. Zespół wyróżniony tytułem „Mistrza Techniki Polskiej — 1973 r.” (od lewej inż. Bogusław Zdunek, inż. Ryszard Krześniak, mrg inż. Marek Konarski) omawia bieżące usprawnienia wprowadzane do pamięci **PT-3**. W wyniku usprawnień powstała nowa wersja zmodernizowanej pamięci pod kryptonimem **PT-3M** (foto CAF)



Rys. 2. **MERAMAT** wykorzystuje najnowsze zdobycze technologii. Na zdjęciu monter układów elektronicznych Elżbieta Wesółowska w czasie sprawdzania połączeń ramy kaset wykonywanych metodą owijania (foto CAF)



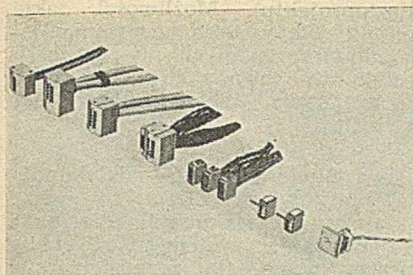
Rys. 3. Połączenia logiczne sprawdzane są za pomocą automatycznego testera zaprojektowanego i wykonanego przez specjalistów **MERAMATU**. Na zdjęciu monter urządzeń elektronicznych Bożena Popiel w czasie podłączania do testera kasy *interface'u* — czyli zespołu elektronicznego, który zabezpiecza współpracę pamięci **PT-3** z maszyną cyfrową (foto CAF)

ODRA 1304, 1305, 1325, jak również są przeznaczone do współpracy z maszynami **Jednolitego Systemu RIAD**.

Oceną wysokiej jakości **PT-3** było przyznanie złotego medalu na wystawie **JS EMC** w Moskwie w 1973 r.

MERAMAT bardzo szybko — bo w ciągu 1,5 roku opanował i uruchomił seryjną produkcję pamięci **PT-3**. Zespół kierujący tymi pracami uzyskał tytuł „Mistrza Techniki Polskiej — 1973”.

Głowice magnetyczne, oparte na ferrytach gęstych, są również poważną pozycją w produkcji **MERAMATU**. Stanowią one przedmiot eksportu do krajów socjalistycznych i kapitalistycznych, w tym do Stanów Zjednoczonych. Między innymi **MERAMAT** dostarcza głowice znanej firmie włoskiej **OLIVETTI**. Produkuje się głowice do zapisu/odczytu



Rys. 4. Głowice magnetyczne produkowane w MERAMACIE

na taśmie magnetycznej (typ GPT-2 i GPT-3) i głowice typu GL-5 do pamięci bębnowych, oparte na polskich wynalazkach.

Konstrukcje te są stale rozwijane.

Załoga MERAMATU jest młoda, rekrutująca się z zasadniczych szkół zawodowych, średnich szkół technicznych i wyższych uczelni. W związku z planami uruchamiania nowej produkcji autonomicznych urządzeń do przygotowywania i wstępnego przetwarzania danych,

MERAMAT stoi przed problemem skupienia i wykształcenia pracowników do tych nowych zadań. Oprócz bowiem klasycznych inżynierów elektroników, zakład pilnie potrzebuje ludzi do opracowywania systemów operacyjnych, programów w językach ASSEMBLER i COBOL, tworzenia pakietów, programów użytkowych, jak również innych prac związanych z bezpośrednią obsługą przyszłych odbiorców.

Dorota Prawdzic

Czas terażniejszy i problemy przyszłości ZETO Wrocław

Zapewne pojawią się niedługo obszerniejsze monografie opisujące powstanie, 10-letnią historię i dorobek wrocławskiego ZETO. Nim to wszakże nastąpi, w naszym cyklu reportaży o działalności ośrodków informatyki, zaprezentujemy ZETO Wrocław z perspektywy dnia dzisiejszego.

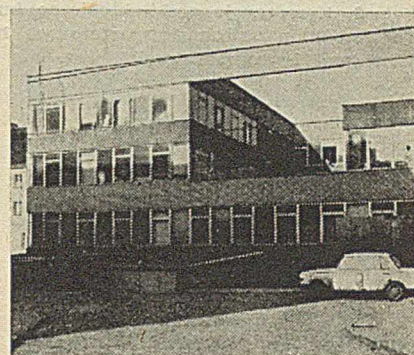
Nie jest to zadanie łatwe, gdyż trzeba mówić właściwie o pięciu ośrodkach: cztery z nich leżą poza granicami miasta, a dwa poza granicami województwa. A więc bohater naszego reportażu — to prawdziwy kombinat, w skład którego — obok ośrodka wrocławskiego —

wchodzi na prawach oddziałów Zakłady w Opolu, Zielonej Górze, Swidnicy i Jeleniej Górze.

Skoncentrujemy się jednak na zakładzie wrocławskim.

Trudno orzec, na ile w wyniku indywidualnych inicjatyw, a na ile w wyniku sprzyjającego klimatu — ale właśnie na odzyskanym Dolnym Śląsku ulokował się pierwszy przemysłowy zakład produkcji maszyn cyfrowych i powstał pierwszy Zakład Elektronicznej Techniki Obliczeniowej.

Czy znaczącą jest tu cecha młodych organizmów gospodarczych — cha-



ZETO — Wrocław

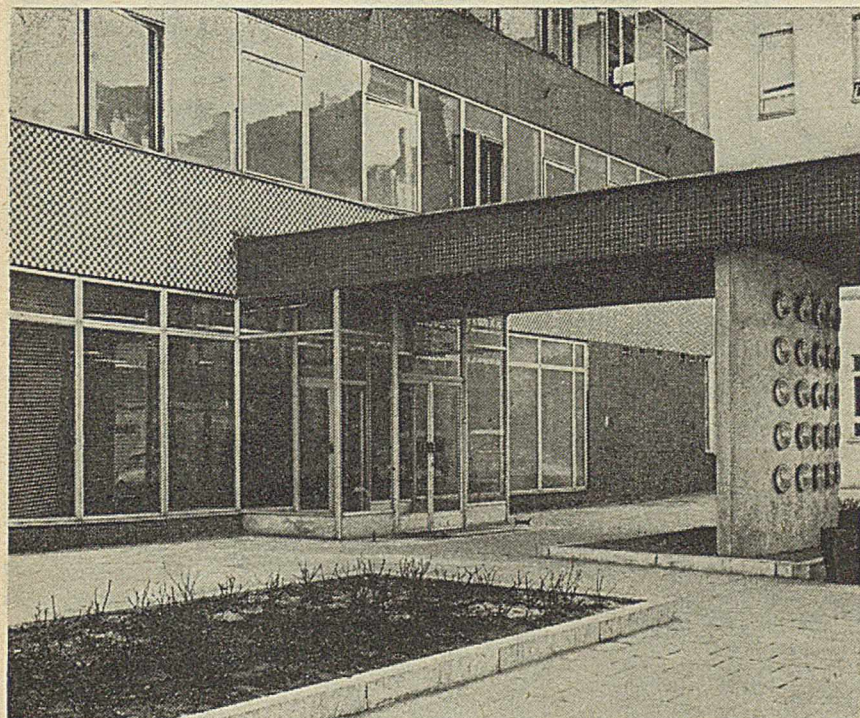


Foto 1 i 2. Budynek wrocławskiego ZETO jest efektowną wizytówką architektury stolicy Dolnego Śląska (prezentujemy fronton i „plecy” budynku)

rakterystyczna dla nich wrażliwość na innowacje?

Przypadający na lata sześćdziesiąte dynamiczny rozwój informatyki zbiegł się z wielką aktywizacją przemysłową ziem nad Odrą. PA-FAWAG, DOLMEL i inne zakłady przemysłowe dyktowały tempo. Informatyka stwarzała tu szansę na lepszą sprawniejszą organizację, podsuwała nowe rozwiązania. Przemysł chciał to wykorzystać. Silny ośrodek obliczeniowy stał się potrzebą chwili. W tym okresie rozrastają się Zakłady ELWRO, pilnie poszukując partnera do próbnego eksploataowania produkowanego sprzętu. Najchętniej — w pobliżu. Powstaje więc ZETO.

Dziś, ZETO-Wrocław oferuje dla blisko 200 przedsiębiorstw wedle potrzeby do wyboru pełne lub fragmentaryczne cykle usług. Pełny cykl — to opracowanie projektu systemu, oprogramowanie, przygotowanie maszynowych nośników informacji i praca komputera. Odbiorcami pełnych cykli usługowych są duże jednostki gospodarcze, które zamierzają w bliższym lub dalszym czasie uruchomić własne zakładowe ośrodki informatyki. Zanim to nastąpi, zakłady te — czynnie

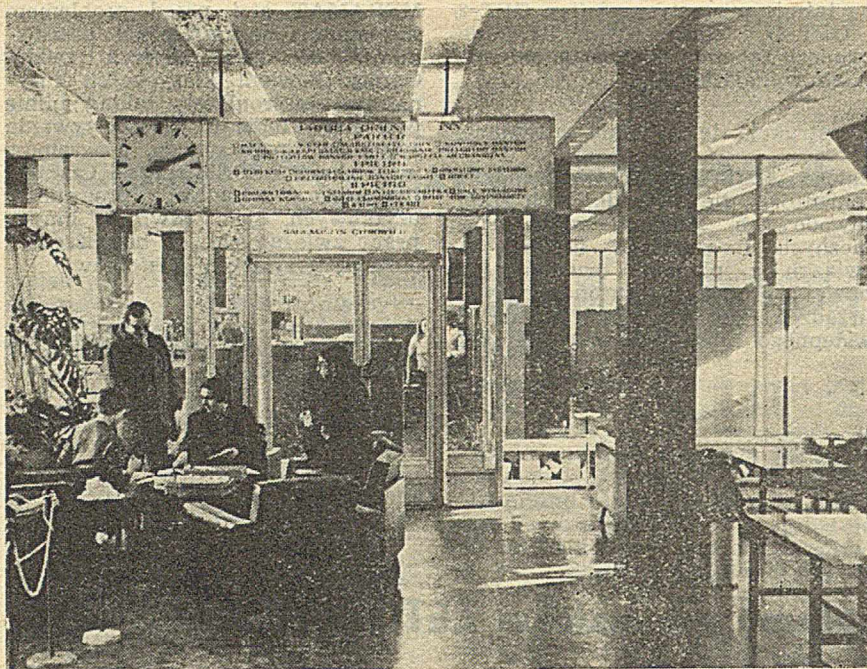


Foto 3. Kompozycję architektoniczną budynku znakomicie wspomaga architektura wnętrza. W przestronnym parterowym hollu, klienci ZETO konsultują się z projektantami i programistami

korzystając z pomocy ZETO, nabywają niezbędnego do organizacji własnego ośrodka doświadczenia, a przy okazji szkolą kadre. Taką koncepcję wybrał np. DOLMEL, który po wieloletnim korzystaniu z usług ZETO, z początkiem tego roku uruchomił własny ośrodek. Małe jednostki gospodarcze korzystają z reguły fragmentarycznie ze standardowych programów bibliotecyjnych ZETO.

Wśród użytkowników korzystających z pełnego cyklu usług zdarzają się i tacy, którzy nie zamierzają uruchomić własnego ośrodka i upatrują korzyści we wzmacnianiu mocy obliczeniowej ZETO; w takim przypadku rozbudowa potencjału ZETO odbywa się w oparciu o środki inwestycyjne tych zakładów, które stają się udziałowcami lub współwłaścicielami sprzętu. Obecnie spora część wartości środków trwałych ZETO, szacowanych na 210 mln zł, przypada na udziały przedsiębiorstw partycypujących.

Niezależnie od usług w zakresie systemów przetwarzania danych, zakład wrocławski prowadzi na zlecenie różnych instytucji szkolenie projektantów, programistów i operatorów. W 1973 r. przeszkolono na kursach 380 projektantów, 120 programistów i 70 przedstawicieli kadry kierowniczej z przedsiębiorstw, będących klientami ZETO. W latach 1969—73 przeszkolono ogółem na różnego typu kursach 2126 osób.

Można też wymienić inne rodzaje usług, takie jak konserwacja i regeneracja taśm magnetycznych, instalacja urządzeń klimatyzacyjnych, projektowanie budowy ośrodków obliczeniowych (przy współpracy

Miastoprojektu). Możliwości takie wyłoniły się w toku narastania doświadczeń osiąganych przy pracach podejmowanych właściwie na rzecz macierzystego Zakładu przez jego służby.

Niewątpliwie jednak — działalnością podstawową ZETO są usługi z zakresu przetwarzania danych i projektowania systemów. Systemy opracowywane dla przedsiębiorstw — użytkowników dałoby się podzielić na cztery zasadnicze typy: systemy technicznego przygotowania produkcji, systemy prowadzenia i kontroli gospodarki materiałowej, systemy planowania produkcji, systemy gospodarowania kadrami. Z wymienionych wyżej systemów korzystają zazwyczaj przedsiębiorstwa o seryjnej produkcji. Realizacja systemów obiektowych, które opracowuje głównie mgr Lesław Wolański, absorbuje większość zatrudnionych w ZETO projektantów. Wspomniane w pobieżny sposób zadania realizuje się w oparciu o dwa komputery MIŃSK 22 oraz jedne z pierwszych egzemplarzy produkowanych w ELWRO komputerów ODRA 1304 i 1305. W sumie ZETO Wrocław ma zainstalowane

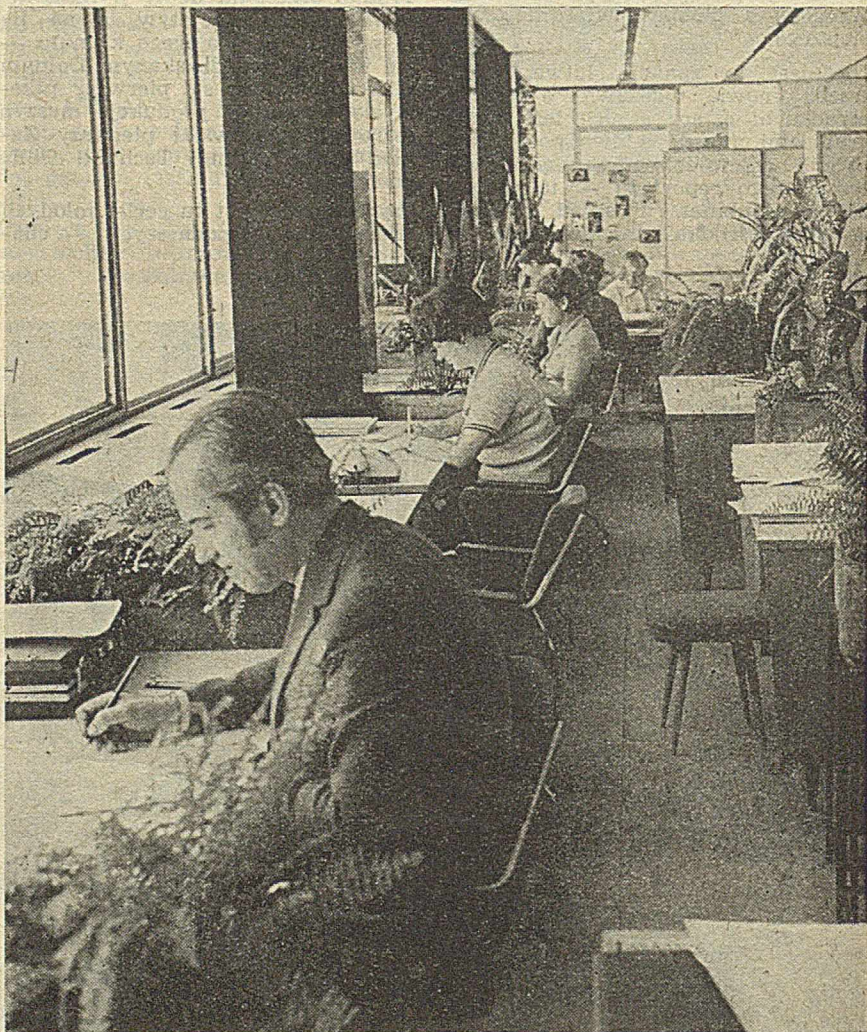


Foto 4. Jak w ogrodzie. W takiej atmosferze nawet gdy jest stosunkowo tłoczno, pracuje się przyjemniej. Na pierwszym planie projektant mgr Ryszard Sadowski

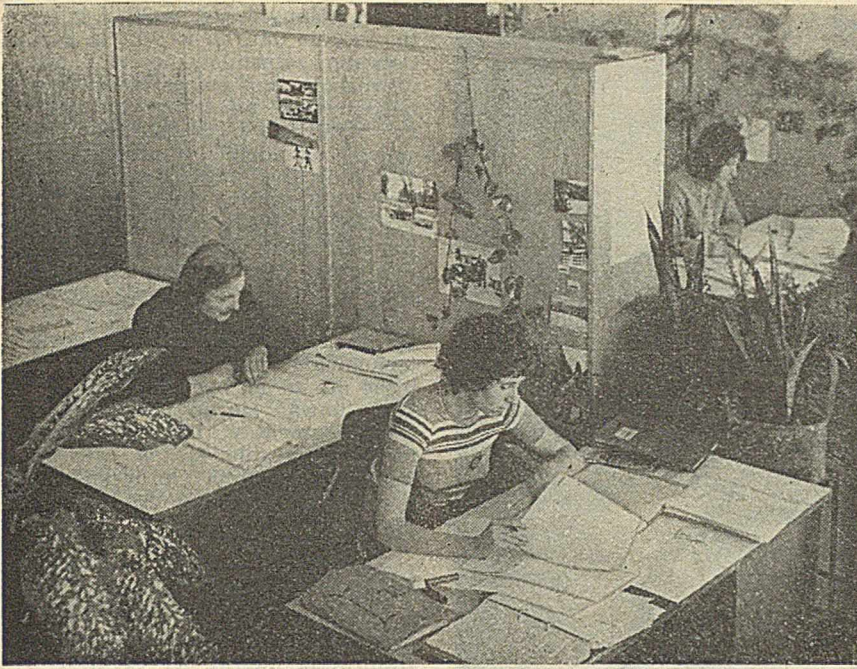


Foto 5. Szybkie rozpakowywanie programów ICL jest warunkiem zwiększenia uniwersalności maszyn serii ODRA. Pracują nad tym programiści: Teresa Kopczyńska i Piotr Kasperski

cztery komputery we Wrocławiu, dwa w Jeleniej Górze, po jednym — w Zielonej Górze i Opolu. Nad najbardziej chyba spektakularnym systemem pracuje obecnie zespół mgr inż. Jerzego Piotrowskiego. Jest to realizowany przy współ-

pracy ELWRO i Politechniki Wrocławskiej wielodostępny system POLRAX-2, oparty na komputerze głównym ODRA 1305 w konfiguracji dyskowej, działający w trybie abonenskim. Przewiduje się, że w pierwszym etapie służyć on będzie



Foto 7. Wdrożenie systemu POLRAX-2 powinno zapoczątkować usystematyzowanie współpracy z użytkownikami. Nad tym zagadnieniem pracuje intensywnie zespół pracowników systemów cyfrowych pod kierownictwem mgr inż. Jerzego Piotrowskiego

15 użytkowników. Warunkiem uruchomienia POLRAX-2 jest wdrożenie systemu operacyjnego GEORGE-3. Nastąpić to powinno w II połowie 1974 r., przy czym nadmienić należy, że wrocławskiemu ZETO z pomocą we wdrażaniu GEORGE-3 przychodzą doświadczenia ośrodków ZETO-Gdynia i Ośrodka Elektronicznego GUS, które pracują równolegle nad tym problemem. Współpraca z ZETO Gdynia rozciąga się również na wymianę doświadczeń z wdrażania systemów organizacji produkcji — NIMMS, PROMPT i FLUTO.

Mimo okazałej liczby maszyn — Ośrodek wrocławski pracuje na trzy zmiany. Świadczy to o dużym popycie na usługi informatyczne. Dyrektor ZETO, mgr Jerzy Trybalski, musi stosować selekcję



Foto 6. Przy pulpicie ODRY 1304 operator Bożena Kamińska

reflektantów, biorąc pod uwagę ciężar gatunkowy ich zadań dla gospodarki narodowej i stopień opanowania problematyki informatycznej. Wybiera się przede wszystkim przedsiębiorstwa, które potrafią jasno określić swoje potrzeby oraz właściwie przygotować dane do przetwarzania, względnie dobrze sformułować zadania do rozwiązania.

Stale zwiększający się popyt na usługi informatyczne jest istotną determinantą dla rozwoju ZETO. W 1974 r. we wszystkich ośrodkach ZETO wchodzących w skład wrocławskiego przedsiębiorstwa zainstalowane zostaną komputery ODRA 1305. Dąży się do usprawnienia procesu przygotowywania danych przez dwójakie rozwiązania: z jednej strony przez rozwój sekcji przygotowywania danych u użytkowników; z drugiej strony —

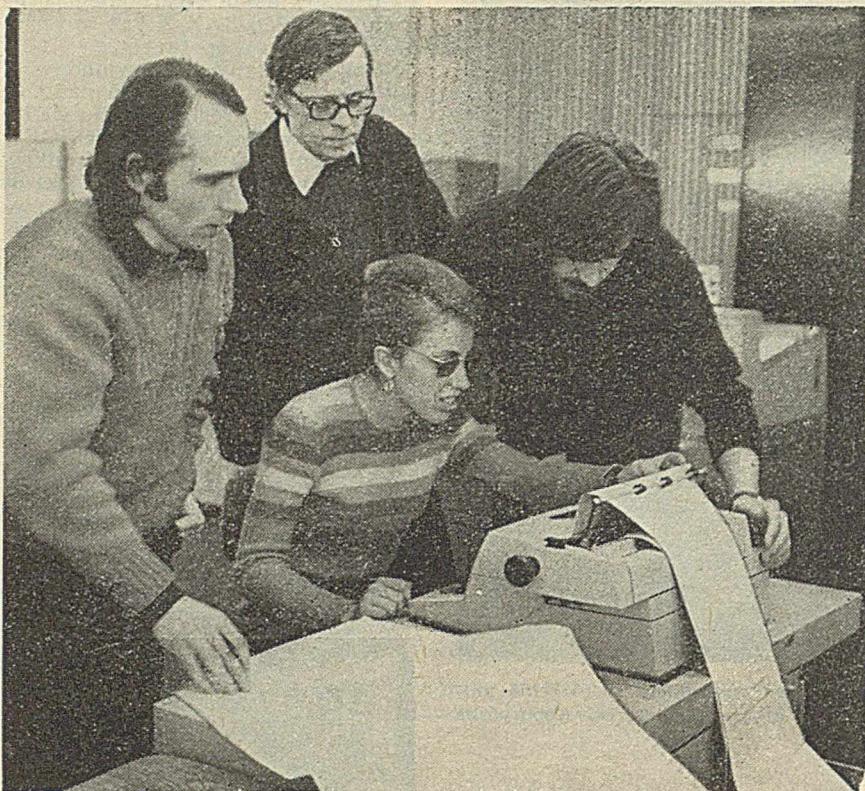


Foto 8. Warunkiem wstępnym do eksploatacji systemu POLRAX-2 jest wdrożenie systemu operacyjnego GEORGE-3. Pierwsze próby z tym systemem na ODRZE 1305 przeprowadzają projektanci: Jerzy Płotrowski, Andrzej Mierzwiak i Edward Bieleńnik oraz st. operator Krystyna Zielińska



Foto 9. Dyrektor ZETO mgr Jerzy Trybułski: „W sytuacji gdy popyt na elektroniczne przetwarzanie przerósł możliwości ZETO, selekcja użytkowników stała się koniecznością. Pracujemy i tak na 3 zmiany a wartość naszych usług w 1973 r. wyniosła 105 mln złotych”

przez zmiany w technologii przygotowywania danych w samym ZETO. Ostatnio zainstalowano tam cztery urządzenia do nanoszenia danych bezpośrednio na taśmy magnetyczne. Myśli się o przekazaniu większym zakładom minikomputerów DATA-POINT 2200, dwóch egzemplarzy RC-3600 i dwóch egzemplarzy MERA 302, które są w posiadaniu ośrodka wrocławskiego. Będą one służyć jako urządzenia do koncentracji i przekazywania danych.

Zmiany też doczekać się powinna technologia projektowania. Zamiast pisać dla zadań specjalne programy, konstruować się je będzie z gotowych segmentów. Do realizacji tego pomysłu powoła się w tym roku Zakład Produkcji Oprogramowania pod kierunkiem mgr Andrzeja Ramuła.

Należy sądzić, że powyższe zamierzenia zostaną zrealizowane, zwłaszcza, że całe przedsiębiorstwo wrocławskie dysponuje liczną, wykwalifikowaną kadrą: 800 osób, w tym 300 projektantów i programistów, jest zatrudnionych w całym ZETO; w ośrodku wrocławskim pracuje 460 osób, w tym 50 programistów pod kierownictwem mgr Michała Nekara. Dużą pomocą dla ZETO-Wrocław jest współpraca z silnymi ośrodkami wyższych uczelni wrocławskich: Uniwersytetu, Politechniki i Wyższej Szkoły Ekonomicznej. Wiele korzyści obustronnych daje bliska współpraca z ELWRO: ZETO otrzymuje szybko maszyny do pracy, a jednocześnie stanowi dla Zakładów ELWRO swego rodzaju poligon dla sprawdzania nowych maszyn z rodziny ODRA, jak również testowania i rozpakowywania oprogramowania firmowego ICL.

Krystyn Bernatowicz
Foto CAF

Seminarium i prezentacja MITRA 15 i IRIS 80 we Wrocławiu

W dniach od 18 do 28 lutego br. odbyło się we Wrocławiu seminarium poświęcone prezentacji minikomputera MITRA 15 oraz systemu IRIS 80. Organizatorami byli UNIWERSYTET WROCŁAWSKI oraz francuska firma COMPAGNIE INTERNATIONALE POUR L'INFORMATIQUE.

W programie seminarium przewidziano wykłady, pokazy sprzętu, indywidualne rozmowy z przedstawicielami CII oraz możliwość pracy z wykorzystaniem prezentowanego sprzętu, w tym również realizacji własnych programów obliczeniowych i innych. Założenia te w zasadzie zrealizowano.

Wykłady były prowadzone przez wybitnych specjalistów CII. Podzielono je na dwie grupy tematyczne; w pierwszej znalazły się zagadnienia dotyczące minikomputera MITRA 15, w drugiej problemy związane z systemami informatycznymi zawierającymi duży komputer IRIS 80.

W ramach pierwszego cyklu tematycznego przedstawiono dane techniczne maszyny MITRA 15 oraz współpracujących z nią urządzeń peryferyjnych, typowe konfiguracje, systemy operacyjne ze szczególnym uwzględnieniem SGF 15, programy użytkowe oraz język dydaktyczny LSE. Dużo czasu poświęcono prezentacji typowych zastosowań minikomputera MITRA 15.

W drugim cyklu przedstawiono dane techniczne maszyny IRIS 80, typowe konfiguracje ze szczególnym uwzględnieniem pracy z transmisją danych na duże odległości, programy użytkowe oraz systemy: operacyjny SPIS 8, przekazywania informacji STRATEGE i organizacji banku danych SOCRATE.

W trakcie pokazów wykorzystywano minikomputer MITRA 15 współpracujący z:

- pamięcią dyskową typu 15201 o pojemności 400 kbajtów
- czytnikiem kart typu 72125 o szybkości 300 kart/min.
- czytnikiem (perforatorem taśm) typu 15062 o szybkości czytania 300 znaków/s. i perforacji 60 znaków/s.
- drukarką wierszową typu 15410 o szybkości 400 wierszy/min.
- 3 monitorami ekranowymi IRISCOPE 200 o pojemności 640 znaków każdy.
- oraz dalekopisem o szybkości 10 znaków/s.

Pokazano realizację kilku programów, między innymi, pozwalających wykonywać różnego typu obliczenia matematyczne i uzyskiwać na ekranie graficzne przedstawienia rozwiązań, robić obliczenia kalendarzowe, porządkować elementy zbiorów według założonego kryterium, prowadzić grę na liczbach generowanych losowo. Uczestnikom semi-

narium umożliwiono realizację własnych programów oraz doraźne ułożenie krótkich programów w języku LSE i uruchomienie ich.

W drugim tygodniu próbowano dokonać połączenia minikomputera MITRA 15 z komputerem IRIS 80 umieszczonym w ośrodku obliczeniowym w Louvenciennes pod Paryżem. W takiej konfiguracji MITRA 15 miał pełnić rolę koncentratora umożliwiającego uzyskiwanie informacji, przegrupowywanie danych i ich teletransmisję, a również obliczenia wstępne i obliczenia na danych uzyskanych z dużej maszyny. Połączenie z Louvenciennes nie doszło do skutku, przy czym jako powód podano zły stan techniczny łączy telefonicznych we Wrocławiu.

Uczestnicy seminarium otrzymali komplety prospektów dotyczących prezentowanego sprzętu. Ponadto w czasie konsultacji indywidualnych mieli możliwość uzyskania dodatkowych informacji i wyjaśnienia ewentualnych wątpliwości.

Należy podkreślić dobrą organizację seminarium oraz wysoki poziom wygłoszonych wykładów.

Przedstawiciel COMPAGNIE INTERNATIONALE POUR L'INFORMATIQUE poinformował, że w ciągu najbliższych miesięcy zostanie otwarte w Warszawie stałe przedstawicielstwo handlowe tej firmy.

KOMUNIKAT

V Letnia Szkoła Metod Numerycznych i Informatyki

● Zakład Zastosowań Matematyki i Metod Numerycznych Instytutu Matematyki Uniwersytetu Gdańskiego, przy współpracy z Instytutem Maszyn Matematycznych Uniwersytetu Warszawskiego i Instytutem Matematycznym Uniwersytetu Wrocławskiego, organizuje w czasie 19.VIII—31.VIII.1974 r. kolejną dwutygodniową V Letnią Szkołę Metod Numerycznych i Informatyki. Celem Szkoły jest intensyfikacja szkolenia i doskonalenie kadr zatrudnionych w szkolenictwie wyższym i poza nim, w zakresie objętym problematyką Szkoły.

● Komitet Organizacyjny Szkoły zaakceptował następujący zespół wykładów:

- 1) Dr Krzysztof Moszyński: Zagadnienia brzegowe i własne dla równań różniczkowych zwyczajnych — 8 godz.
- 2) Dr Edward Neuman: Zastosowanie funkcji sklepanych w aproksymacji — 6 godz.
- 3) Mgr Aleksander Janicki: Zastosowanie funkcji sklepanych do rozwiązywania równań różniczkowych — 6 godz.

4) Mgr Stanisław Lewanowicz: Liniowe równania różnicowe — 8 godz.

5) Prof. dr Adam Rybarski, Doc. dr Roman Zuber: O pewnym zagadnieniu brzegowym z ruchomym brzegiem w teorii filtracji nieustalonej — 10 godz.

6) Mgr Helena Krupicka, Mgr Ewa Wilhelm: Doświadczenia z prowadzenia pracowni programowania na Uniwersytecie Wrocławskim — 6 godz.

7) Doc. dr H. Schwetlick (NRD): Równania nieliniowe i minimalizacja funkcjonalów (w języku angielskim) — 8 godz.

8) Mgr Michał Jankowski: Złożoność obliczeniowa i równoległość w analizie numerycznej — 6 godz.

9) Dr J. Wendt (NRD): Metody rozwiązywania równań różniczkowych cząstkowych — 4 godz.

10) Dr Jerzy Loska: COBOL, 8 godz.

11) Mgr Stanisław Szpakowicz, Dr Stanisław Waligórski: Systemy konwersacyjne — 6 godz.

12) Dr Michał Tessarowicz: FORTRAN — 8 godz.

13) Dr Michał Tessarowicz: Systemy operacyjne — 8 godz.

14) Dr Ryszard Zieliński: Metody Monte Carlo w zadaniach optymalizacji — 6 godz.

15) Dr Gerhard Paulin (NRD): Kompilery systemów wielodostępnych — 4 godz.

● Poza wykładami organizatorzy przewidują możliwość wygłoszenia przez uczestników Szkoły pewnej liczby komunikatów (20—30 min.).

● Zajęcia Szkoły odbywać się będą w pomieszczeniach dydaktycznych Uniwersytetu Gdańskiego w Oliwie.

● Zgłoszenia uczestników Szkoły należy nadsyłać do 30 maja 1974 r. na adres: V Letnia Szkoła Metod Numerycznych i Informatyki, Instytut Matematyki UG, 80-264 Gdańsk — Wrzeszcz, ul. Marchlewskiego 16a.

● Komitet Organizacyjny zastrzega sobie prawo dokonania pewnych zmian w podanym wykazie wykładów.

Komitet Organizacyjny Szkoły prosi również o nadsyłanie wniosków, opinii i uwag mogących przyczynić się do zwiększenia efektywności Szkoły.



Przedstawiamy firmę BENSON

Francuska firma BENSON jest jednym z największych światowych producentów urządzeń automatyzacji projektowania. Powstała w r. 1955 w Stanach Zjednoczonych a jej oddział we Francji miał za zadanie jedynie sprzedaż wytworzonego w Ameryce sprzętu. Wyraźne usamodzielnianie się przedstawicielstwa francuskiego firmy datuje się od r. 1960, a od roku 1968 BENSON jest samodzielną firmą francuską.

Zakłady BENSONA znajdują się w podparyskiej miejscowości Creteil. Jest to fabryka o powierzchni 5 km² z pomieszczeniami biurowymi i handlowymi (2,7 km²), zatrudniająca 170 pracowników.

Ostatnie lata były dla tego przedsiębiorstwa okresem gwałtownego rozwoju. Obroty BENSONA wzrosły z 2,8 M franków w 1965 r. do 13,4 M franków w roku 1971. Ostatnie, aktualne dane, specjalnie nadesłane na naszą prośbę przez centralę paryską, wynoszą: obrót w 1972 roku 15 M franków, w 1973 — 19 M fr, w tym eksport 6 M fr. Wg oceny firmy należy do niej aktualnie 80% francuskiego rynku maszyn infograficznych i 10% rynku światowego. Firma sprzedawała wiele urządzeń do krajów skandynawskich, Holandii, Belgii, Szwajcarii, Włoch, Portugalii i Hiszpanii. Pierwsza filia utworzona w RFN z siedzibą w Wiesbaden, działa już od 1971 roku. W tej chwili BENSON czyni starania o wejście na rynek japoński i amerykański.

Firma ma bardzo ożywione kontakty z krajami socjalistycznymi, gdzie zainstalowano 71 spośród 650 zainstalowanych na świecie maszyn autokreślarskich. Firma może poszczycić się faktem, że spośród kilkudziesięciu maszyn zainstalowanych w ZSRR — pięć pracuje w centrum naukowym w Nowosybirsku.

W dn. 12—17 listopada 1973 r. w Moskwie wielkim zainteresowaniem cieszyła się samodzielna wystawa wyrobów BENSONA (sprzedano większość wystawionych maszyn, a w PRAWDZIE ukazała się pochlebna wzmianka o wystawie).

W Polsce, według stanu na koniec lutego 1974 r., pracuje 12 maszyn BENSONA (w tym 7 maszyn autokreślarskich): 6 w Warszawie, 2 w Krakowie, 2 w Toruniu oraz po 1 w Bydgoszczy i Sosnowcu.

Firma BENSON wchodzi w skład Grupy Tranchant Electronique (ogółem 7 fabryk wytwarzających podzespoły, aparaturę elektroniczną i urządzenia mechaniczne).

Przedstawicielstwo firmy (a zarazem przedstawicielstwo Grupy Tranchant Electronique) mieści się w Warszawie przy ul. Szpitalnej 1

m 36. W ubiegłym roku złożył wizytę w naszym kraju naczelny dyrektor firmy BENSON — inż. Jean Mourier — na zaproszenie Stowarzyszenia Inżynierów i Techników Budownictwa. Tranchant Electronique S.A. Biuro Techniczne w Warszawie zapewnia m.in. konserwację i serwis techniczny maszyn BENSONA. Urządzenia automatyzacji projektowania wytwarzane przez Firmę BENSON są typowymi urządzeniami peryferyjnymi mogącymi pracować z każdą maszyną cyfrową. Sprzęt tego typu cieszy się obecnie na całym świecie coraz większym zainteresowaniem. BENSON produkuje pełny zestaw maszyn infograficznych — autokreślarki, drukarki katodowe, czynni-

rostowych — BENSON wprowadził ostatnio do produkcji nowe serie maszyn infograficznych opracowanych według najnowocześniejszych koncepcji, a mianowicie:

— autokreślarki z wbudowanym interpolatorem liniowym o jeszcze większej szybkości rysowania, dające rysunek o wyższej jakości oraz znaczną oszczędność na czasie pracy komputera

— używane w zarządzaniu drukarki katodowe na mikrofilmie lub na mikrofiszach z modułami graficznymi o szybkości drukowania dziesięciokrotnie wyższej od szybkości najlepszych klasycznych drukarek dla elektronicznych maszyn cyfrowych.

Oto kilka parametrów tych maszyn: — autokreślarka TD 222: wymiary rysunku 84 × 120 cm; liczba pisaków 4; maksymalna szybkość rysowania wzdłuż osi 7,5 cm/sek, wzdłuż przekątnej — 10,6 cm/sek; wbudowany interpolator liniowy; wymiary zewnętrzne: długość 185 cm, szerokość 135 cm, wysokość 105 cm

— drukarka katodowa COM 310: drukarka: 3 do 3,5 klatek na sekundę, tj. 15.000 wierszy na minutę, 64 wiersze na stronę, 135 znaków w wierszu; 64 znaków alfanumerycznych; moduł graficzny w reżimie przyrostowym; kamera 16—35 i 105 mm, wybór różnych stopni redukcji przy pomocy wymienionych obiektywów 24 x, 42 x lub 48 x, licznik klatek; licznik fisz, rejestracja znaków odniesienia; wymiary zewnętrzne: długość 200 cm, wysokość 171 cm, głębokość 64 cm

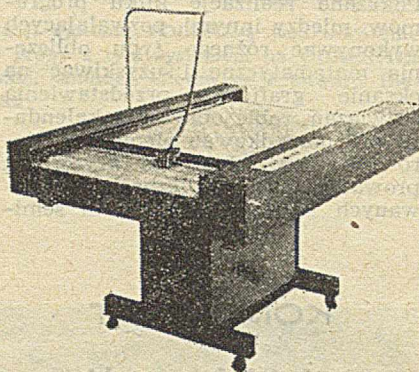
W zależności od potrzeb użytkownika maszyny BENSONA mogą pracować:

— bezpośrednio (*on-line*), ze wszystkimi typami maszyn cyfrowych. Dla naszych krajowych zastosowań ważne jest tu, że maszyny BENSONA są przystosowane do pracy z komputerami ODR 1200 i 1300 oraz z komputerami Jednolitego Systemu

— za pośrednictwem transmisyjnych linii telefonicznych (telerysowanie)

— za pośrednictwem taśmy magnetycznej (czytniki taśmy magnetycznej BENSONA, 7 i 9 ścieżek, 556/300/1600 bpi) lub taśmy dziurkowej (czytniki produkcji MERA — BŁONIE).

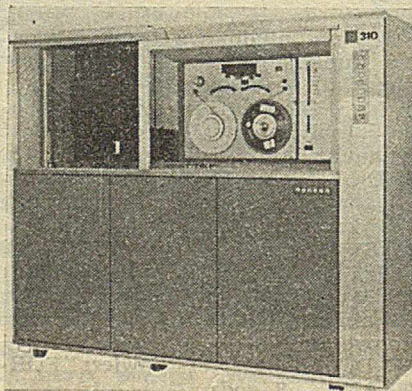
Warszawskie przedstawicielstwo BENSONA organizuje dla użytkowników kursy eksploatacji i programowania maszyn BENSONA. Ostatnio kurs taki odbył się w dniach 11—13 lutego, przy współudziale Instytutu Dróg i Mostów Politechniki Warszawskiej.



Rys. 1. Autokreślarka TD 222

ki punktów, krzywych i filmów (zapis liczbowy informacji graficznej).

Stałe dążenie do nowych rozwiązań technicznych powoduje, że — po szybkich autokreślarkach przy-



Rys. 2. Drukarka katodowa COM 310

Marek Holyński

Japońskie Centrum Informatyki opublikowało ostatnio sprawozdanie na temat bieżącej sytuacji komputerów elektronicznych w Japonii w postaci tzw. „Białej Księgi o Komputerach, 1973”. W końcu marca 1973 r. japońskie systemy komputerowe o przeznaczeniu ogólnym obejmowały łącznie 17.255 maszyn wartości 1.373.200 mln jenów, czyli \$ 4577 mln (w cenach sprzedaży). Toteż pod względem liczby komputerów Japonia uplasowała się na drugim miejscu, za Stanami Zjednoczonymi, które posiadają 57 730 maszyn wartości 26 100 mln \$. W końcu września 1973 r. zainstalowanych w Japonii było ogółem 9607 sztuk komputerów produkcji krajowej (wartości 685 500 mln jenów czyli \$ 2285) i 5199 sztuk zagranicznych wartości 576 300 mln jenów, czyli \$ 1.921 mln.). Procentowo maszyny japońskie stanowiły 64,9% ogólnej liczby kom-

puterów w roku finansowym 1970, do 243 mln jenów w roku finansowym 1972. Jeśli chodzi o usługi, to więcej niż połowa, bo 54,7% usług, wyświadczonych zostało w dziedzinie kalkulacji biurowych (46,7%). Japonia pozostaje więc wciąż w tyle za Stanami Zjednoczonymi, jeśli chodzi o zastosowanie komputerów w dziedzinie przetwarzania danych.

Zbyt japońskich przedsiębiorstw programujących wynosi jedynie jedną czwartą zbytu takich samych przedsiębiorstw amerykańskich. Toteż Japonia jest wciąż jeszcze nieprzygotowana do sprzedaży usług z dziedziny oprogramowania. (Instytucje programujące przygotowują i sprzedają opracowania techniczne potrzebne do korzystania z komputerów).

Jest sprawą pilną, aby tę „lukę w oprogramowaniu” szybko wypełnić tym bardziej, że rząd japoński zde-

r. gwałtownie się rozwinął i obecnie może się pochwalić roczną produkcją wartości jena 382.200 mln czyli \$ 1.274 mln., (łącznie z urządzeniami zewnętrznymi).

Tym niemniej opracowanie nowych systemów komputerowych wymaga ogromnych inwestycji, długich badań i opracowań, które nie mogą być podejmowane przez instytucje prywatne. Toteż rząd przeznaczył sumę około 10.000 mln jenów czyli 33 mln \$ na lata 1966—1971 na opracowanie „sprawnie działających komputerów elektronicznych”.

W roku 1971 rząd zainwestował dodatkowo 35.000 mln jenów (116 mln \$) na opracowanie „wzorcowego systemu przetwarzania danych” do roku finansowego 1978-go.

Plan ten — podzielony na trzy etapy — zakłada podstawowe badania oraz opracowanie elementów i materiałów w latach 1971—74, próbną produkcję modeli urządzeń do pisania, sporządzania wykresów i głośników oraz sprawdzenie tych modeli w latach 1973—76, jak też produkcję i próby całego systemu w latach 1974—78. Plan ten został przyjęty. Należy się spodziewać, że postęp w ewolucji społeczeństwa obsługiwanego przez komputery sprawi, iż w latach 1980-tych trzeba będzie zastosować nowy system komputerów, zdolny do bezpośredniego wchodzenia, rozpoznawania i przetwarzania informacji w postaci pisma, wykresów, przedmiotów i głosu, czego obecne komputery nie są w stanie efektywnie robić.

W celu pobudzenia rozwoju prac nad opracowaniem nowych komputerów i pokrewnych im urządzeń, rząd japoński utworzył system subsydiów, który ma się przyczynić do stworzenia nowych konkurencyjnych systemów i przyspieszenia rozwoju różnego rodzaju urządzeń peryferyjnych.

W tym systemie subsydiów ogółem 34.000 mln jena czyli 113 mln \$ zostanie wypłaconych w okresie ponad trzech lat, począwszy od roku finansowego 1972.

Główni użytkownicy komputerów w Japonii

Urzędy finansowe	197.100 mln jenów	= 657 mln \$
producenci maszyn elektronicznych	172.000 mln	= 573 mln \$
handel hurtowy i detaliczny	116.000 mln	= 387 mln \$
urzędy państwowe	90.900 mln	= 303 mln \$
producenci maszyn transportowych	769.100 mln	= 263 mln \$
usługi	76.500 mln	= 255 mln \$
przemysł żelaza i stali	62.700 mln	= 209 mln \$
przemysł chemiczny i petrochemiczny	54.600 mln	= 182 mln \$
komunikacja, łączność i środki masowego przekazu	44.100 mln	= 147 mln \$
rząd	43.400 mln	= 144 mln \$

puterów, importowane — 35,1%. Wartość komputerów japońskich (54,3%) wynosiła prawie tyle samo, co zagranicznych (45,7%).

Przetwarzanie danych

1 stycznia 1973 r. japoński przemysł przetwarzania danych liczył 541 przedsiębiorstw o 710 ośrodkach przetwarzania i wykazywał wciąż tendencje rozwojowe. Przeciętny roczny dochód takiego przedsiębiorstwa zwiększył się ze 180 mln

cydował się zliberalizować od marca 1976 r. bezpośrednio inwestycje zagraniczne w krajowym przemyśle oprogramowania. Podobne kroki zostaną podjęte w przemyśle komputerowym od początku roku finansowego 1974 (kwiecień 1974 — marzec 1975).

Japońska polityka komputerowa

Japoński przemysł komputerowy, który zaczynał od produkcji komputerów na użytek handlu, w 1957

Normowanie zużycia materiałów

W Związku Radzieckim przystąpiono do opracowania ogólnopanstwowego zautomatyzowanego systemu normowania zużycia zasobów materiałowych. System przewiduje dynamiczną aktualizację tych norm, uwzględniając wpływ postępu technicznego. Obliczenia norm materiałowych, wraz z wprowadzeniem do nich zmian, scalaniem, tworzeniem zagregowanych wskaźników techniczno-ekonomicznych niezbędnych dla różnych szczebli planowania i zarządzania — będzie się przeprowadzać za pomocą komputerów. Ogniwami projektowanego systemu będą przedsiębiorstwa produkcyjne, ministerstwa branżowe i

urzędy centralne, w skali ogólnopanstwowej i poszczególnych republik. Celem systemu jest dokonywanie syntezy opracowań pochodzących z wielu podsystemów.

Oczekuje się, że dzięki temu systemowi uporządkuje się dokumentację źródłową w zakładach produkcyjnych, usprawni się przygotowanie i przetwarzanie danych, zaś instytucje planujące uzyskają oparte na sprawdzonych informacjach normy zużycia poszczególnych grup materiałowych. Umożliwi to polepszenie techniczno-ekonomicznego uzasadnienia planów, szersze wprowadzanie postępowych norm

materiałowych i zmniejszenie prac administracyjnych. Jeden z takich branżowych podsystemów normowania materiałów działa już od 1973 r. w Ministerstwie Budowy Przychodów, Środków Automatyzacji i Systemów Sterowania ZSRR.

W ośrodku obliczeniowym tego Ministerstwa, na podstawie istniejących norm, wylicza się zapotrzebowanie na materiały, oddzielnie dla każdego przedsiębiorstwa i łącznie dla całej branży. Obecnie wprowadza się operatywną aktualizację norm i obliczenie średnio ważonych grupowych norm zużycia najważniejszych materiałów.

ETOCHEM

Mały System Informatyczny Centrali Zjednoczenia

Wprowadzenie kompleksowych zasad ekonomiczno-finansowych do systemu zarządzania Wielkimi Organizacjami Gospodarczymi (WOG) w resorcie przemysłu chemicznego postawiło nowe zadania przed służbą informatyczną resortu w ogóle, a WOG w szczególności. Dotychczasowy system zarządzania za pomocą szczegółowych wielkości i wskaźników dyrektywnych zastąpiono w nowych „Kompleksowych zasadach ekonomiczno-finansowych” metodą zarządzania za pomocą ograniczonej liczby wskaźników syntetycznych.

Głównym wskaźnikiem efektywności ekonomicznej działalności jednostki gospodarczej jest dochód, którego podstawowymi elementami są sprzedaż i koszty materialne. Rozwiązanie takie zwiększa liczbę stopni swobody w działalności optymalizacyjnej i decyzyjnej na każdym szczeblu zarządzania.

Węzłowym punktem systemu, z punktu widzenia zakresu uprawnień i dostępnych narzędzi polityki ekonomicznej jest centrala WOG. Zarówno na etapie planowania wieloletniego, przy ustalaniu optymalnego wariantu reprodukcji rozszerzonej, jak i przy ustalaniu najkorzystniejszej alokacji produkcji, a następnie oddziaływaniu na jej realizację, WOG musi dysponować wysoko wydajnym systemem informacyjnym, zdolnym nie tylko gromadzić i uogólniać informacje potrzebne WOG; ale również zapewniać niezbędną optymalizację przedsięwzięć i wariantowość decyzji. Bazę techniczną do realizacji takiego systemu może stworzyć jedynie informatyka.

W pełnej świadomości potrzeby takiego systemu kierownictwo Zjednoczenia Przemysłu Organicznego w Warszawie zainicjowało, we współpracy z Resortowym Ośrodkiem Informatyki i Doradztwa Organizacyjnego ETOCHEM, opracowanie takiego systemu.

Cel i zakres systemu informatycznego

Określenie celu, a tym bardziej zakresu docelowego systemu informatycznego dla potrzeb centrali WOG jest aktualnie bardzo trudne, ze względu na początkową fazę realizacji nowego systemu zarządzania. Nie byłoby chyba również celowe szerokie stosowanie informatyki tam, gdzie nie jest ona do tej pory znana od strony praktycznej.

Z tych to właśnie względów z całości zagadnień wydzielono na początek tylko pewne podsystemy informatyczne, które są najbardziej do tej pory określone a jednocześnie obejmują najważniejsze problemy z punktu widzenia zakresu działalności WOG. Są to zagadnienia sprzedaży, kosztów, zdolności produkcyjnej, bilansów materiałowo-towarowych, kadry. Tworzą one względnie odosobnione moduły problemowe systemu których projektowanie przebiega sekwencyjnie.

Mały System Informatyczny Centrali Zjednoczenia (MSICZ), dysponując danymi o ustalonym wyżej zakresie przedmiotowym ze wszystkich przedsiębiorstw produkcyjnych podległych zjednoczeniu, ma zapewnić dostarczanie operatywnych informacji niezbędnych do okresowej i doraźnej oceny sytuacji na odcinkach objętych systemem w przedsiębiorstwach i zjednoczeniu. W momencie planowania będą to wariantowe bilanse materiałowo-towarowe (moduł 3), zaś w okresie realizacji planów — dane o realizacji sprzedaży i przewidywanym dochodzie (moduł 1) oraz ocena stopnia wykorzystania zdolności produkcyjnych (moduł 2).

W zakresie danych o sprzedaży system będzie miał możliwość wyeliminowania części sprawozdawczości operatywnej. W warunkach WOG szczególną uwagę przywiązuje się również do należytego gospodarowania wysokowykwalifi-

kowaną kadrami inżyniersko-techniczną i kierowniczą. Dlatego też w systemie MSICZ będzie prowadzić się operatywną ewidencję tej kadry (moduł 4).

Podstawowym użytkownikiem systemu będzie kierownictwo centrali WOG, ale na żądanie informacje będą również dostarczane przedsiębiorstwom i kierownictwu resortu.

Organizacja MSICZ

System MSICZ będzie się opierał na dwóch podstawowych zbiorach informacyjnych: normatywnym i operatywnym.

Zbiór normatywny składa się z trzech kartotek:

- kartoteki indeksowej wyrobów
- kartoteki indeksowej podstawowych surowców
- kartoteki zdolności produkcyjnych aparatury.

Zbiór operatywny natomiast obejmuje:

- kartotekę stanów i obrotów wyrobów gotowych
- kartotekę wysokokwalifikowanej kadry.

● Kartoteka indeksowa wyrobów gromadzi dane o stosowanej symbolice, nazewnictwie i normach wyrobu, dane cenowe i kosztowe, a także normy zużycia podstawowych surowców na produkcję jednostki kalkulacyjnej wyrobu.

● Kartoteka surowców zawiera dane o stosowanej symbolice, nazewnictwie, normach technicznych i cenach surowców.

● Kartoteka zdolności produkcyjnych zawiera informacje o efektywnych zdolnościach produkcyjnych aparatury w przeliczeniu na produkty podstawowe wykonywane na tej aparaturze.

● Kartoteka stanów i obrotów produkcji towarowej zawiera informacje o przychodach z produkcji i innych kierunków oraz o rozchodach na sprzedaż (eksport wg

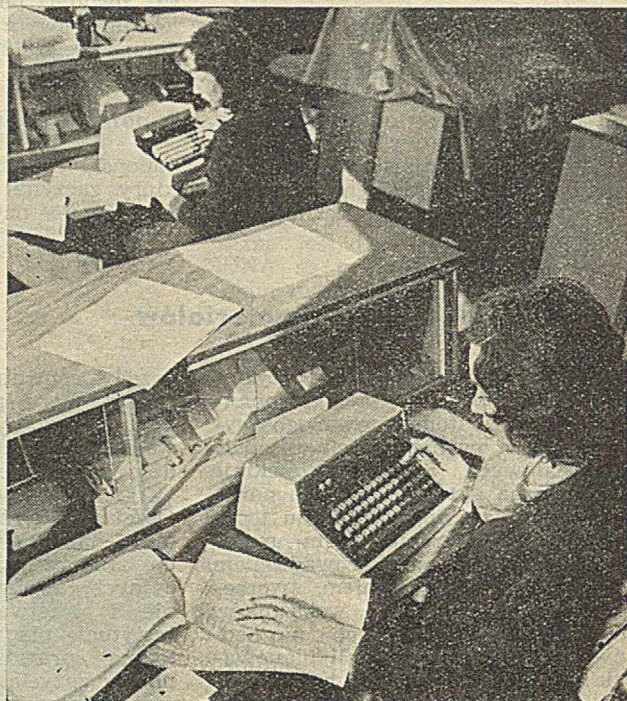


Foto 1. Przygotowywanie danych w Ośrodku ETOCHEM (CAF)

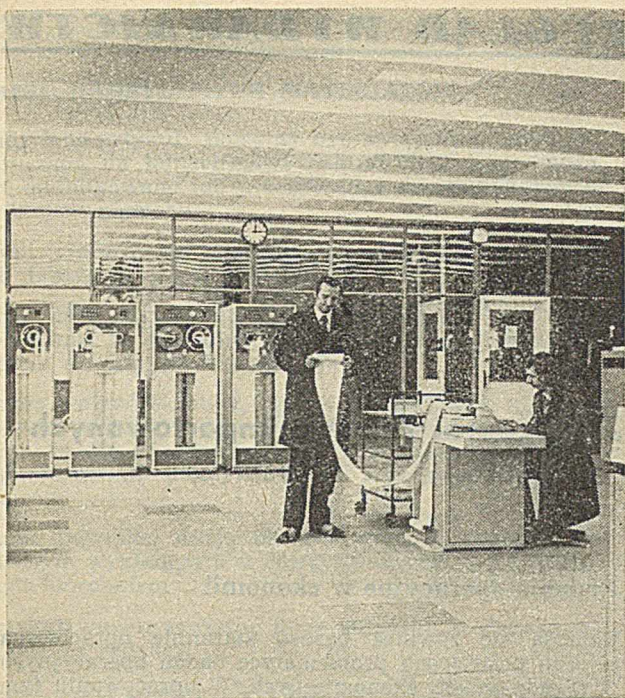


Foto 2. Fragment sali z komputerem ODRA 1305 (CAF)

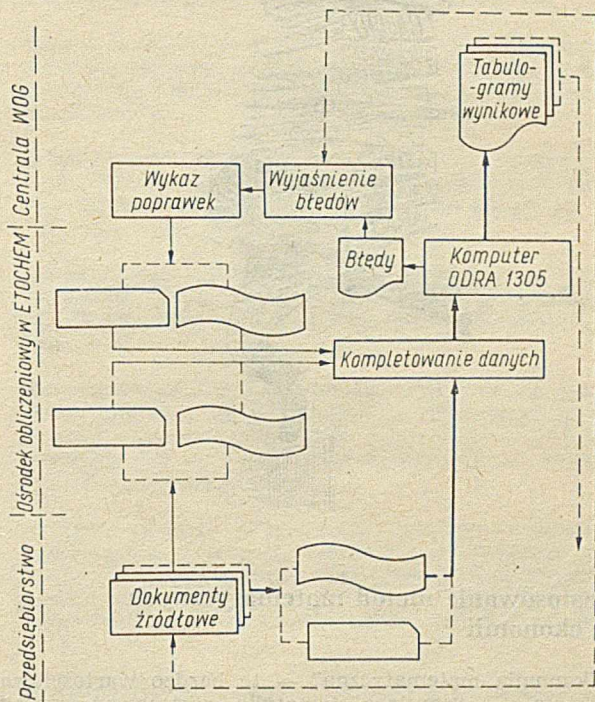
obszarów i kraj wg kierunków) i na pozostałe cele w ujęciu ilościowo-wartościowym, a także informacje organizacyjne służące do kontroli kompletności i wiarygodności informacji źródłowej.

● Kartoteka kadrowa zawiera informacje głównie o przygotowaniu zawodowym, specjalistycznym i ogólnym wysokokwalifikowanej kadry inżyniersko-technicznej i kierowniczej.

Zbiory informacyjne będzie się aktualizować w cyklach wynikających z częstości korzystania z tych zbiorów. I tak kartoteka indeksowa wyrobów oraz kartoteka stanów obrotów będą aktualizowane z częstością 2-3 dni. Pozostałe kartoteki będą aktualizowane raz w miesiącu (kwartalnie). Informacja aktualizująca ma spływać do ośrodka obliczeniowego bezpośrednio z przedsiębiorstw wchodzących w skład WOG, z tym, że karty indeksowe, zasilające zbiory normatywne będą przechodzić przez branżowe przedsiębiorstwo obrotu, które dokonuje weryfikacji tych kart i nadaje indeks w przypadku wprowadzenia nowych pozycji do kartotek. Zbiór operatywny (stany i obroty wyrobów) zasilą się głównie fakturami krajowymi, wystawionymi w przedsiębiorstwach WOG za pomocą maszyn fakturujących z przystawką dziurkującą taśmę papierową (w przypadku ich awarii można dziurkować karty lub przysłać dokumenty) oraz fakturami eksportowymi wystawianymi w przedsiębiorstwach handlu zagranicznego. Ze względu na dużą rozbieżność momentów czasowych wydania i fakturowania dostaw na eksport, w systemie uwzględnia się ponadto dowody Wz dotyczące eksportu.

W celu zapewnienia nadzoru nad gospodarką zapasami wyrobów i nad powiązaniem z modułem systemu dotyczącym wykorzystania zdolności produkcyjnych, wprowadza się dodatkowo jedyne w systemie zestawienie zbiorcze wystawiane w przedsiębiorstwach, mianowicie zestawienie przychodów z produkcji i innych kierunków oraz rozchodów na inne cele (poza sprzedaż). System uwzględnia zarówno dokumenty podstawowe, jak i korekty ogólnie stosowane w praktyce.

Przetwarzanie danych odbywać się będzie na komputerze ODRA 1305 zlokalizowanym w ETOCHEMIE. W wyniku eksploatacji systemu w zakresie modułu 1, tj. danych o realizacji sprzedaży i dochodzie będzie się uzyskiwać 13 zestawień okresowych dotyczących wykonania sprzedaży, prawidłowości stanów zapasów, opłacalności sprzedaży itp. Ponadto na żądanie, początkowo telefonicznie, a po uruchomieniu teletransmisji bezpośrednio użytkownicy będą mogli otrzymywać zestawienia dodatkowe.



Ideowy schemat przepływu informacji w systemie MSICZ obrazuje rys. 1.

Etapy prac

Nakreślony zakres prac realizowany jest etapowo.

W pierwszej połowie 1974 r. wprowadza się moduł 1; moduły 2 i 4 przewiduje się wprowadzić w połowie 1975 r.; do końca 1975 r. planuje się wprowadzenie ostatniego modułu 3.

Na bazie systemu MSICZ przewiduje się opracowanie modeli symulacyjnych do optymalizacji planu produkcji oraz systemu wariantowania decyzji.

System MSICZ z jednej strony zapełni luki w zapewnieniu dopływu informacji dla kierownictwa WOG na jednym z podstawowych odcinków jego zainteresowań, a z drugiej strony stworzy kierownictwu możliwość bezpośredniego operowania systemem informatycznym. Stanowi to etap przygotowawczy do wprowadzenia bardziej rozwiniętych systemów, bez których działanie WOG, a tym bardziej działanie optymalne, będzie niemożliwe.

Zbigniew Bieńko
ETOCHEM



Zastosowanie metod matematycznych w ekonomii

„Ekonomia matematyczna” — to bardzo wartościowa pozycja — ujmująca tematykę zastosowań metod matematycznych w ekonomii — opracowana przez profesora K. Lancastera z Uniwersytetu w Kolumbii, tłumaczona na język rosyjski pod kierunkiem profesora D. B. Judina.

W części I uwypukla się istotę dociekań optymalizacyjnych i omawia się szczegółowo ogólne zadanie optymalizacji, teorię programowania liniowego, klasyczne i współczesne metody rozwiązywania takich problemów.

W części II charakteryzuje się modele statyczne i omawia się modele typu „nakłady — wyniki”, liniowe i nieliniowe modele optymalizacji oraz problem równowagi ogólnej.

W części III przedstawia się modele dynamiczne i rozpatruje się zagadnienie zbilansowanego wzrostu, problem efektywnego i optymalnego wzrostu oraz zagadnienie stateczności.

Poza tym, w rozdziale IV — w dodatkach 1—11, przedstawia się także podstawy przydatnych tutaj działów matematyki — teorii zbiorów, algebry liniowej, subtelności teorii zbiorów wypukłych i funkcji wypukłych, zagadnienie ciągłych odwzorowań, idee topologii, rachunek wariacyjny i inne tego rodzaju problemy.

W książce tej, obok klasycznych metod, przedstawia się również nowe ujęcia oraz modyfikacje metod już istniejących, co świadczy o dużej wartości tego opracowania. Ponadto świetne komentarze profesora D. B. Judina — w tłumaczeniu książki na język rosyjski — dodatkowo podnoszą walory tej pracy.

*) K. Lancaster — *Matematičeskaja ekonomika (Ekonomia matematyczna)*. Izd. „Sovietskoje radio”, Moskwa 1972, s. 464 (tłum. z ang.).

Z księgarni książek importowanych

Badania operacyjne w ekonomii

Ukazała się również bardzo starannie opracowana książka poświęcona problematyce badań operacyjnych w zagadnieniach ekonomicznych. W opracowaniu tym uwypukla się sprawę modeli standardowych przydatnych dla tego rodzaju dociekań.

W rozdziale 1 omawia się rachunek macierzowy oraz możliwości jego zastosowania w ekonomii, w rozdziale 2 — modele prób w badaniach statystycznych (zastosowanie rachunku prawdopodobieństwa, zmienne losowe i ich rozkłady, próby i ich ocenę, analizę korelacji i regresji, statystyczną kontrolę jakości), w rozdziale 3 — teorię obsługi (strukturę systemu obsługi, określenie celu w tych badaniach, specjalne systemy obsługi), w rozdziale 4 — teorię odnowy (w odniesieniu do systemów technicznych), w rozdziale 5 — modele zapasów (deterministyczne i stochastyczne), w rozdziale 6 — modele sieciowe (technikę konstruowania sieci oraz swoiste metody badań), w rozdziale 7 — programowanie liniowe (matematyczną formalizację problemu, metodę geometryczną i metodę simpleks, wraz z jej wariantami, problem dualności, zagadnienie zmiennych z dolnym i górnym ograniczeniem, programowanie całkowitobowe, programowanie parametryczne, problem nieliniowości oraz problem optymalizacji wielocelowej), w rozdziale 8 — problem transportowy (matematyczną formalizację problemu, metody rozwiązywania problemów transportowych, niezbilansowane zadanie transportowe, problemy z ograniczoną przepustowością tras, uogólniony problem transportowy, wielorodzajowy i wielostopniowy model transportowy, ekonomiczną interpretację zmiennych dualnych, a także problem przyporządkowania), w rozdziale 9 — planowanie przebiegu i problem kolejności (charakteryzuje się zagadnienie i rozpatruje się modele odzwierciedlające konkretne sytuacje występujące w praktyce), w rozdziale 13 — matematyczne narzędzia analizy rynku (obliczanie trendów, ustalanie wahań okresowych, zastosowanie analizy regresji) i wreszcie — w rozdziale 14 omawia się także problem symulacji, z uwzględnieniem zarówno modeli deterministycznych, jak i stochastycznych.

Książka powyższa powinna być przydatna przede wszystkim dla kadry kierowniczej zarządzającej naszą gospodarką narodową, jak również dla studentów wyższych uczelni ekonomicznych i inżynierijno-ekonomicznych.

Jan Adamus

**) *Matematische Standardmodelle der Operations-forschung (Matematyczne standardowe modele badań operacyjnych)* — Autoronkollektiv. Verlag Die Wirtschaft, Berlin 1972, s. 719.

Bibliografia wydawnictw polskich z dziedziny informatyki

● **COBOL** — KURZYDEM E. i inni, PWE, Warszawa 1973, ss. 246, cena zł 19,—

Zalety i wady języka COBOL, historia jego powstania. Podstawowe wiadomości dotyczące EMC i ich programowania.

Reguły gramatyczne języka COBOL, Hierarchia danych. Pole elementarne i grupowe. Format pozycji pola. Przemieszczanie danych w pamięci operacyjnej. Arytmetyka w języku COBOL. Grupowe dodawanie, odejmowanie, przenoszenie. Wyrażenie warunkowe. Struktura części proceduralnej. Warunkowa zmiana sterowania. Powtarzalność elementów. Drukowanie danych. Format pozycji zbioru. Operacje wejścia-wyjścia. Kontrola realizacji programu. Operacje na wybranych znakach pola. Identyfikacja programu. Pojęcie modułu. Ćwiczenia końcowe.

Dodatki: wykaz słów zastrzeżonych języka COBOL, formaty języka COBOL, programy przykładowe. Książka zawiera zbiór wykładów przedstawionych metodą programowanego nauczania z podstaw programowania w języku COBOL.

Przeznaczona ona jest dla programistów, analityków i projektantów systemów EPD oraz kadry kierowniczej użytkowników systemu EPD.

● **Komputer w budownictwie. Praktyczne zastosowanie informatyki i badań operacyjnych** — HUSARSKI K. ARKADY, Warszawa 1973, ss. 135, cena zł 20,—

Informatyka, jej podstawowe pojęcie i metody działania. Badania operacyjne: systemy, modele, optymalizacja, bilansowanie, programowanie liniowe i nieliniowe, metody sieci zależności, badania operacyjne problemów specjalistycznych, zastosowanie komputera w projektowaniu. Systemy automatycznego przetwarzania: technologia automatycznego przetwarzania informacji, praktyka wdrażania i eksploatacji systemów API, efektywność stosowania ETO.

Książka przeznaczona jest dla projektantów, kierownictwa i służb ekonomicznych przedsiębiorstw budowlanych.

● **Sprzęt informatyki. Cz. 1: Encyklopedyczne wiadomości o maszynach cyfrowych** — MIEŚCICKI J. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1973, ss. 21. Materiały szkoleniowe dla Telewizyjnego Kursu Informatyki B6.

Algorytm, program, język. Organizacja typowej maszyny cyfrowej. Wieloprogramowanie i wielodostępność. Hardware a software: sprzęt a oprogramowanie. Pokolenie i rodziny.

● **Transmisja danych, organizacja i technika przestrzennego działania systemów informatycznych** — FIJAŁKOWSKI W. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1973, ss. 35. Materiały szkoleniowe dla Telewizyjnego Kursu Informatyki B7.

Transmisja danych w systemach informatycznych. Proces przesyłania danych. Błędy i zabezpieczenie przed błędami. Wykorzystanie sieci telexowych i telefonicznych. Współpraca komputera z odległą stacją abonencką. Powszechnie sieci komputerowe.

● **Projektowanie zautomatyzowanych systemów zarządzania** — HANUSZ T. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1973, ss. 43. Materiały szkoleniowe dla Telewizyjnego Kursu Informatyki B,8/9.

Struktura projektu zautomatyzowanego systemu zarządzania. Struktura zautomatyzowanego systemu zarządzania. Przebieg projektowania systemów EPD.

Rodzaje zautomatyzowanych systemów zarządzania, oraz zasady ich projektowania. Metody projektowania zautomatyzowanych systemów zarządzania.

● **Zasady programowania komputerów** — KUCHARSKI D. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1973, ss. 20. Materiały szkoleniowe dla Telewizyjnego Kursu Informatyki B 10/11.

Cechy szczególne komputera. Program, programowanie. Proces programowania. Analiza zadania. Sieć działań programowania. Pisanie programu. Tłumaczenie programu na program. Wyniki. Uruchamianie, testowanie i przygotowanie programu do eksploatacji. Przykładowy program.

● **Automatyzacja obliczeń zawodowych inżynierskich** — MIROWSKI W. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1973, ss. 15. Materiały szkoleniowe dla Telewizyjnego Kursu Informatyki C12.

Pojęcia podstawowe. Przykłady zastosowań. Systemy automatyzacji procesu projektowania (SAP). Sprzęt informatyczny. Spodziewane efekty zastosowania informatyki w projektowaniu.

● **Kierowanie produkcją samochodów** — PLUCIŃSKI S. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1973, ss. 22. Materiały szkoleniowe dla Telewizyjnego Kursu Informatyki D 16.

Charakterystyka przedsiębiorstwa. Informacja w procesie kierowania. Proces wytwarzania. Podsystem informatyczny kierowania produkcją. Organizacja służb planowania i kontrola wykonania produkcji.

● **Przygotowanie organizacyjne użytkowników dla wprowadzenia systemów informatycznych** — PLUCIŃSKI S. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1973, ss. 19. Materiały szkoleniowe dla Telewizyjnego Kursu Informatyki E 21.

Założenia wstępne. Przygotowanie warunków organizacyjnych: analiza struktury organizacyjnej i przepływu informacji, kodowanie danych, przystosowanie dokumentacji do przetwarzania, utworzenie Ośrodka Przetwarzania Danych. Przygotowanie Kadr. Przygotowanie warunków technicznych.

● **Przygotowanie organizacyjne WPTO w Bydgoszczy do wdrożenia podsystemu EWIDOS** — PEWIŃSKA Z. Wyd. Towarzystwa Naukowego Organizacji i Kierownictwa, Bydgoszcz 1973, ss. 17. Konferencja naukowa n.t. „Podstawowe problemy organizacji systemów informatycznych w handlu wewnętrznym” Bydgoszcz — kwiecień 1973 r.

Zadania działu EPD, zasięg prac przygotowawczych. Wnioski.

Materiały przeznaczone są dla personelu kierowniczego przedsiębiorstw handlu wewnętrznego.

● **Systemy informatyczne w organizacjach handlu wewnętrznego** — JAKUBOWSKI G. Wyd. Towarzystwa Naukowego Organizacji i Kierownictwa, Bydgoszcz 1973 ss. 16. Konferencja naukowa n.t. „Podstawowe problemy organizacji systemów informatycznych w handlu wewnętrznym” Bydgoszcz — kwiecień 1973 r.

Koncepcja resortowego systemu informatycznego w zakresie działalności handlowej (RSI-MHWiU), drugi etap realizacji zastosowań ETO w państwowym handlu wewnętrznym, charakterystyka SEPD-WPTO. Załączniki 3.

Materiały przeznaczone są dla personelu kierowniczego przedsiębiorstw handlu wewnętrznego.

● **Projektowanie technologii zmechanizowanego przetwarzania danych statystycznych** — NABOŹNY T. Wyd. GUS, Warszawa 1973 r., ss. 124, ZMIAOS nr 11.

Czynniki wpływające na efektywność przetwarzania danych za pomocą maszyn licząco-analitycznych.

Treść i organizacja projektowania. Zasady projektowania wzorów formularzy statystycznych. Opracowanie dokumentacji technologicznej: zapoznanie się z tematem opracowania, zasady projektowania maszynowych nośników danych, zasady projektowania nośników informacji wynikowej, dokumentacja procesu technologicznego. Dokumentacja technologiczna przetwarzania danych spisu drzew i krzewów owocowych.



Zakład Doświadczalny
Instytutu Chemii Fizycznej i Instytutu Chemii Organicznej
POLSKIEJ AKADEMII NAUK
Warszawa, ul. Kasprzaka 44

CYJANOPAN B-4

Klej cyjanoakrylowy skleja w ciągu sekund.

- Klei: metale, ceramikę, szkło, tworzywa termoplastyczne, tworzywa termoutwardzalne, drewno i wiele innych materiałów — wytrzymałość 100 do 300kG/cm².
- Przy sklejaniu nie wymaga utwardzaczy, temperatury, ciśnienia ani żadnych przyrządów pomocniczych.
- Niezbędny element w nowoczesnej technologii produkcji i montażu wszelkiego rodzaju urządzeń w przemyśle precyzyjnym, w przemyśle urządzeń pomiarowych, w elektronice oraz elektrotechnice.
- Kleje cyjanakrylowe stosowane przez wszystkich czołowych producentów maszyn, wyrobów elektrotechnicznych i elektronicznych na świecie są obecnie produkowane w Polsce przez Chemipan pod nazwą Cyjanopan B-4.

Dystrybutorem kleju Cyjanopan B-4 jest POCH Gliwice, ul. Sowińskiego 11.

Zamówienia należy składać w Wojewódzkich Hurtowniach Wyrobów Przemysłu Chemicznego.

Cena 20-gramowego opakowania wynosi 212 zł.

WCT/912/K/74



ZETO-Wrocław dysponuje urządzeniami do atestacji i regeneracji taśm magnetycznych. Stacja umożliwia lokalizację i usuwanie uszkodzeń, które powstają na taśmach w czasie eksploatacji.

Stacja jest wyposażona w dwa podstawowe urządzenia:

- Certifler model 300
- Cleoner/Rewinder (2 szt.)

Korzystając ze stacji daje gwarancję prawidłowej eksploatacji taśmoteki. Zakład prowadzi również usługi na zewnątrz. Koszty regeneracji kształtują się w granicach od 1/3 do 1/20 kosztów zakupu nowej taśmy.

Zamówienie należy kierować pod adresem:

ZAKŁAD ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ,
DZIAŁ PRZETWARZANIA DANYCH

Tel. 454-31 do 37, ul. Ofiar Oświęcimskich 7-13, 50-069 Wrocław

Testowanie taśm magnetycznych



WCT/848/K/74