P. 1877/76

1 1976

informatyka

SPIS TREŚCI NR 1/76

		str.
	STEP 74 jako oprogramowanie narzędziowe o zastosowaniu przemysłowym — *Andrzej Brandt*	1
	Czym są i co dają profile programów — *Stanisław Matwin, Marek Missala*	6
	Rejestry buforowe — *Henryk Więcek*	9
	Koncepcja modyfikacji standardowego systemu organizacji bazy danych „Bill of material processor" (BOMP) — *Marek Zdziubany*	11
	Zautomatyzowany system zarządzania kombinatem LEUNA-WERKE WALTER ULBRICHT — *Werner Kunitz*	13
	Skuteczność metod ochrony zbiorów informacji w systemach informatycznych — *Andrzej Sokołowski*	16
INFORMATYKA NA UCZELNIACH	Jakie będą komputery lat osiemdziesiątych? Prognozy rozwoju sprzętu — tłum. i oprac. *A. Kojemski*	19
KADRY INFORMATYKI	Laboratorium programowania współbieżnego (1) — *Jerzy Chmurzyński*	23
ETO W ZARZĄDZANIU	Nauczanie matematyki na poziomie podstawowym przy użyciu języka APL — *Andrzej J. K. Plewicki*	25
Z KRAJU	Organizacyjne problemy zastosowania ETO w zarządzaniu — *Andrzej Aranowski*	26
	Niezupełnie a jednak w cieniu — *Krystyn Bernatowicz*	29
ZE ŚWIATA	Informacje różne	32
	Ewolucja parku urządzeń do wprowadzania danych w krajach Europy zachodniej	33
	Informacje różne	34
ZE ZJEDNOCZENIA INFORMATYKI OŚRODKI INFORMATYKI PREZENTUJĄ	Informacje różne	35
	Informatyka w Fabryce Sprzętu Samochodowego w Łodzi. Zastosowania — perspektywy rozwoju — *Jerzy Arnold*	36
KONCEPCJE INFORMATYKI TERMINOLOGIA	Co wynika z przeszłości — *MAT*	37
	Terminologia — *Danuta Kołacka, Anna Wieczorek, Jan Wierzbowski, Julian Winiewski*	39
PROBLEMATYKA BAZY DANYCH	Uwagi do koncepcji Uniwersalnego Systemu Zarządzania Bazą Danych dla komputerów JS EMC — *Witold Jamontt*	40
	Od autorów — *Witold Staniszkis, Stanisław Wężyk*	42
SZTUKA KOMPUTEROWA PRZEGLĄD WYDAWNICTW	Komputery w plastyce (2) — *Marek Hołyński*	44
	Recenzja książki E. I. Lowe'a, A. E. Hiddena: Maszyny cyfrowe w automatyce przemysłowej — *Zbigniew Poznański*	45
	Bibliografia wydawnictw polskich z dziedziny informatyki — oprac. *J. Klamborowski*	22 i 47
	Ogłoszenia	46, 48, III i IV okł.

WYDAWNICTWA CZASOPISM TECHNICZNYCH NOT
Warszawa
Czackiego 3/5

KOLEGIUM REDAKCYJNE

Redaktor Naczelny prof. dr hab. Leon ŁUKASZEWICZ

mgr Krystyn BERNATOWICZ, prof. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), dr inż. Marek HOŁYŃSKI, Władysław KLEPACZ (zasa. redaktora naczelnego) doc. dr hab. Antoni MAZURKIEWICZ, mgr Janusz SPYCHAJ, doc. dr inż. Andrzej TARGOWSKI

Sekretarz Redakcji mgr Krystyna WROŃSKA — Red. tech. Anna POTARSKA

RADA PROGRAMOWA

Mgr inż. Antoni BOSSOWSKI, doc. dr inż. JAN FELICKI, doc. dr inż. Zbigniew GACKOWSKI, doc. dr inż. Aleksander GOLIŃSKI, dr hab. Andrzej GRZYWAK, doc. dr hab. inż. Roman KULESZA, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. inż. Stanisław PASZKOWSKI (wiceprzewodniczący), prof. dr Tadeusz PECHE, mgr inż. Bronisław PIWOWAR, dr inż. Andrzej PLĄSKOWSKI, mgr inż. Tadeusz PODGÓRSKI (wiceprzewodniczący), prof. dr inż. Jerzy SEIDLER, prof. dr inż. Andrzej STRASZAK (przewodniczący), doc. Jerzy TRYBULSKI, dr Tadeusz WALCZAK, prof. dr inż. Stefan WĘGRZYN, dr inż. Jan Z. ŻYDOWO

Redakcja: 00-941 Warszawa, ul. Jasna 14/16, pokój 331, tel. 27-71-40 lub centrala 26-82-61 w. 285, dyżurny redakcji 10,00—13,00

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka". Z. 2. Zam. 607. Papier druk. sat. IV kl. 61 × 86, obj. 6,75 ark. druk. Nakład 6.500. J-35.

Cena egzemplarza zł 15.— INDEKS 36210/36124 Prenumerata roczna zł 180.—

Informatyka

dawniej Maszyny Matematyczne

zastosowania w gospodarce, technice i nauce

Nr 1
MIESIĘCZNIK
1976
ROK XI
Styczeń

ORGAN MINISTERSTWA NAUKI, SZKOLNICTWA WYŻSZEGO I TECHNIKI I POLSKIEGO KOMITETU AUTOMATYCZNEGO PRZETWARZANIA INFORMACJI NACZELNEJ ORGANIZACJI TECHNICZNEJ

BIBLIOTEKA ŚLĄSKA KATOWICE
P. 1877/76

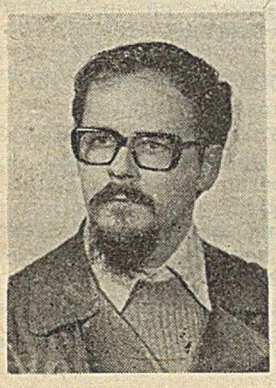
ANDRZEJ BRANDT
Ośrodek Badawczo-Rozwojowy Informatyki
Warszawa

STEP 74
jako oprogramowanie narzędziowe o zastosowaniu przemysłowym

Opisano system STEP 74, który jest zestawem uniwersalnych pakietów programowych wykonujących funkcje zakładania, bieżącej aktualizacji oraz różnorodnego wykorzystywania bazy danych. System przeznaczony jest dla przedsiębiorstw przemysłowych — użytkowników maszyn JS EMC lub IBM serii 360 i 370.

Jednym z narzędzi automatycznego przetwarzania informacji w przedsiębiorstwie, zapewniającym:
— utrzymywania dokładnych i aktualnych zbiorów danych
— szybki dostęp do pojedynczej informacji
jest przedstawiony poniżej zestaw programów organizacyjnych o nazwie:
STEP 74 — System Sterowania Techniczno-Ekonomicznego Produkcją.

System STEP (opracowany przez były Zakład Produkcji Oprogramowania R-POLSYSTEM ZOWAR obecnie OBRI) jest przeznaczony dla polskich przedsiębiorstw przemysłowych — użytkowników Jednolitego Systemu EMC oraz komputerów IBM serii 360 i 370.



Mgr Andrzej BRANDT ukończył Wydział Ekonomiki Produkcji SGPiS w Warszawie (1968). Z tematyką projektowania i wdrażania informatycznych systemów przemysłowych związany jest od 1967: jako projektant, od 1969 — jako kierownik zespołów projektowych (1969—1972 w Zakładach Przemysłu Odzieżowego CORA w Warszawie, 1972—75 w ZETO-ZOWAR, od 1975 w Ośrodku Badawczo-Rozwojowym Informatyki). Jest jednym z autorów systemu STEP.

System STEP charakteryzuje się następującymi podstawowymi cechami:

- strukturą modularną, umożliwiającą tworzenie ze składowych pakietów programowych wersji optymalnej dla konkretnego użytkownika w zależności od jego potrzeb i możliwości
- wyspecjalizowaną bazą danych — jednak z zapewnieniem możliwości zastosowania banku wielobazowego przy użyciu specjalistycznych zbiorów łącznikowych
- możliwością sukcesywnej rozbudowy systemu — bez zasadniczych zmian w strukturze podstawowej bazy danych
- rewelacyjnymi ułatwieniami w obsłudze — dzięki zastosowaniu maksymalnie uproszczonych języków problemowych.

System STEP jest zestawem uniwersalnym pakietów programowych, wykonujących funkcje zakładania, bieżącej aktualizacji oraz różnorodnego wykorzystania wyspecjalizowanej bazy danych konstrukcyjno-technologicznych. Wszystkie pakiety użytkowe, tj. wykorzystujące informacje zgromadzone w bazie danych mogą być dołączane do podstawowych pakietów zakładania i aktualizacji tej bazy w liczbie i kolejności dowolnej, określonej potrzebami i możliwościami użytkownika.

Poszczególne pakiety programowe systemu STEP produkowane są przez **wyspecjalizowane tematycznie grupy projektowo-programowe przy ścisłej współpracy z przedstawicielami przedsiębiorstw przemysłowych** — rzeczywistych i potencjalnych użytkowników systemu. Muszą one spełniać następujące podstawowe warunki:

- maksymalna prostota obsługi

- maksymalne wykorzystanie informacji z podstawowego źródła — bazy danych przy jednoczesnym ograniczaniu do minimum liczby zbiorów uzupełniających
- możliwość aktualizacji jedną informacją wszystkich informacji z nią związanych
- zapewnienie użytkownikowi możliwości wyboru zakresu funkcjonalnego pakietu (realizacja całości lub wybranej części funkcji)
- zapewnienie możliwości samodzielnej rozbudowy zakresu funkcji pakietu dokonywanej przez użytkownika.

Niezależnie od stworzenia możliwości rozbudowy poszczególnych pakietów programowych siłami własnymi użytkownika, zapewniono rozbudowę pakietów oraz całego systemu przez producenta oprogramowania w kierunku realizacji nowych funkcji, na jakie pojawi się zapotrzebowanie.

Tak więc, na etapie implementacji u konkretnego użytkownika, sukcesywna rozbudowa zakresu użytkowego systemu STEP realizowana może być przy użyciu jednej z/lub obu podstawowych metod:

- **zakupu i instalacji** na zestawie komputerowym użytkownika **nowo opracowanych pakietów programowych** wyspecjalizowanych w zakresie funkcji i posiadających wbudowane interfazy z pakietami obsługi podstawowej bazy danych.

Producent oprogramowania oferuje rokrocznie rozszerzony typowy zestaw pakietów programowych tworzący system: STEP 74, STEP 75, STEP 76 itd. Użytkownik może zakupić dowolną wersję, adekwatną do jego potrzeb oraz możliwości wykorzystania; użytkownik posiadający już np. wersję STEP 74 ma możliwość zakupienia i instalacji wszystkich, bądź dowolnie wybranych pakietów programowych, które zostały dodatkowo zastosowane w zestawie typowym 75 i uzyskania tą drogą zestawu STEP 75

- **możliwością uruchamiania siłami własnego zespołu programów** bądź pakietów programowych **realizujących funkcje dodatkowe** i włączania ich do swojej wersji systemu.

Przy założeniu włączania do systemu indywidualnych uzupełnień w miejsce i sposób wskazany przez producenta oprogramowania w dokumentacji opisowej i eksploatacyjnej systemu — użytkownik nie traci prawa do usług gwarancyjnych.

Prostota obsługi systemu STEP zapewniona została dzięki zastosowaniu specjalnych, zwięzłych w opisie **języków programowania dla określonego kręgu problemów, np.:** JOZ — język opisu zbiorów, JOW — język opisu wyjścia (wyników).

Podstawową cechą wymienionych wyżej przykładowo języków problemowych jest możliwość przyswojenia ich **nawet przez użytkownika nie mającego wstępnego przygotowania w zakresie programowania.** Przy zastosowaniu kilku, bądź kilkunastu rodzajów prostych instrukcji, użytkownik jest w stanie sformułować konkretny problem na poziomie projektowym. Problem użytkownika zostaje dzięki działaniu odpowiedniego kompilatora języka podstawowego poddany konwersji na właściwą postać (kompilatory języków problemowych wchodzą w skład pakietów programowych dostarczanych użytkownikowi). W efekcie działania kompilatorów użytkownik otrzymuje gotowy program bądź zestaw programów zdolny wykonać określone, żądane funkcje z zakresu:

- zakładania i aktualizacji podstawowej bazy danych i/lub zbiorów pomocniczych
- wyszukiwania i/lub przetwarzania informacji zgromadzonych w bazie danych i zbiorach pomocniczych
- kontroli poprawności, a w niektórych przypadkach również automatycznej korekty danych wejściowych, danych zgromadzonych w bazie danych oraz wyników
- reorganizacji zbiorów bazy danych i/lub zbiorów pomocniczych.

Zastosowanie języków problemowych umożliwia ponadto bieżącą kontrolę poprawności przebiegu przetwarzania dzięki szeroko rozbudowanej diagnostyce błędów i warunków wyjątkowych. Obsługa niektórych warunków wyjątkowych oraz dokonywania korekt szeregu typowych błędów możliwa jest w sposób automatyczny względnie może być pozostawiona decyzji operatora komputera.

Kompilatory języków problemowych systemu STEP umożliwiają ponadto w trakcie realizacji niektórych problemów, np. zakładania zbiorów, bądź ich reorganizacji, uzyskiwanie standardowych wydruków opisowych stanowiących gotowe elementy dokumentacji technicznej (organizacyjnej i programowej) systemu.

Zakres użytkowo-funkcjonalny systemu

STEP 74 obejmuje następujące zasadnicze funkcje:

- zakładanie podstawowej bazy danych konstrukcyjno-technologicznych (zwanej dalej BAZA DANYCH)
- aktualizację bazy danych
- wyszukiwanie informacji w bazie danych.

Podstawowa baza danych konstrukcyjno-technologicznych zakładana i obsługiwana przez programy systemu STEP 74 obejmuje cztery zbiory kartotekowe umieszczone na dyskach magnetycznych z możliwością bezpośredniego dostępu do informacji.

Zbiory kartotekowe systemu STEP 74:

- KARTOTEKA RODZAJOWA
- KARTOTEKA STRUKTURALNA
- KARTOTEKA TECHNOLOGICZNA
- KARTOTEKA STANOWISK ROBOCZYCH.

KARTOTEKA RODZAJOWA zawiera zespół informacji związanych z produktami finalnymi, zespołami, podzespołami i detalami występującymi w całym asortymencie produkcji przedsiębiorstwa. W kartotece tej można również umieścić materiały podstawowe i pomocnicze zużywane w procesie wytwarzania.

Informacje zawarte w tej kartotece muszą zawierać identyfikator pozycji (części, zespołu, wyrobu) oraz informacje związane z identyfikatorem np. cena, nazwa, jednostka miary itp.

Poza informacjami wprowadzonymi przez użytkownika kartoteka ta posiada zespół informacji adresowych tworzących powiązania z kartotekami: STRUKTURALNĄ i TECHNOLOGICZNĄ.

KARTOTEKA STRUKTURALNA stanowi zbiór informacji mówiących o wzajemnych powiązaniach pozycji umieszczonych w kartotece RODZAJOWEJ i parametrach tych powiązań. Kartoteka ta stanowi odbicie konstrukcyjnej lub technologicznej struktury wyrobu.

Informacjami użytkownika w tej kartotece są parametry typu: norma zużycia detali na zespół, współczynnik braków przy montażu, norma zużycia materiału na detal itp. Pozostała część rekordu zawiera zespół adresów dyskowych umożliwiających powiązania rekordów wewnątrz KARTOTEKI STRUKTURALNEJ oraz powiązanie rekordów KARTOTEKI STRUKTURALNEJ z KARTOTEKĄ RODZAJOWĄ. Jeden rekord dotyczy jednego powiązania. Liczba możliwych powiązań pozycji złożonych ze składnikami (zespołu z wchodzącymi w jego skład częściami) jest praktycznie nieograniczona.

KARTOTEKA TECHNOLOGICZNA grupuje informacje dotyczące sposobu wytworzenia pozycji znajdującej się w KARTOTECE RODZAJOWEJ. Może istnieć nieograniczona liczba rekordów dotyczących jednej pozycji rodzajowej np. rekordów operacji.

W pojedynczych rekordach KARTOTEKI TECHNOLOGICZNEJ znajdują się dowolne informacje związane z operacją, podane przez użytkownika oraz zespół adresów dyskowych niezbędnych dla zachowania powiązań z kartotekami: RODZAJOWĄ i STANOWISK ROBOCZYCH.

Informacją użytkownika musi być identyfikator, np. numer operacji, a mogą być dane dotyczące tej operacji, np. czas jednostkowy wykonania operacji, stawka zaszeregowania roboty, narzędzia itp.

W rekordzie KARTOTEKI TECHNOLOGICZNEJ nie ma informacji o stanowisku, na którym wykonuje się daną operację oraz informacji o części, której dotyczy ten proces technologiczny. Informacje te podawane przy tworzeniu rekordu technologicznego są zamieniane na informacje adresowe do odpowiednich rekordów kartotek: RODZAJOWEJ i STANOWISK ROBOCZYCH.

KARTOTEKA STANOWISK ROBOCZYCH zawiera zbiór informacji dotyczących wszystkich maszyn, urządzeń i innych produkcyjnych stanowisk pracy istniejących w przedsiębiorstwie.

Poza informacjami, np. wartość, parametry technologiczne, fundusz czasu itp., podanymi przez użytkownika, rekord tej kartoteki zawiera adresy wiążące go z KARTOTEKĄ TECHNOLOGICZNĄ. Jeden rekord dotyczy jednego stanowiska (lub grupy stanowisk).

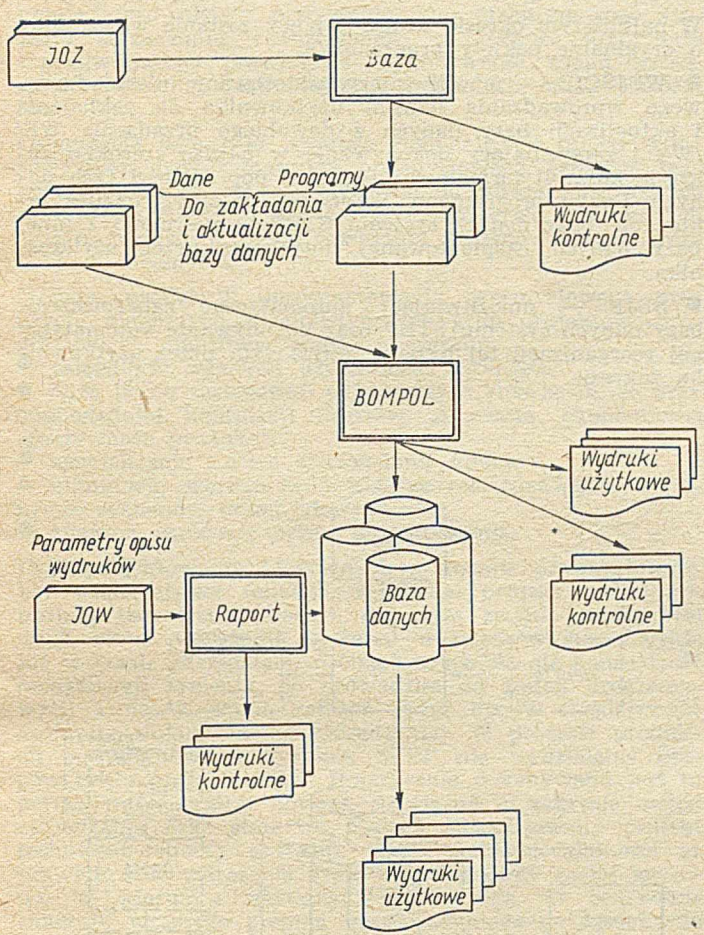
Funkcje zakładania i permanentnej aktualizacji bazy danych realizowane są przez system STEP 74 dzięki zastosowaniu dwu uniwersalnych pakietów programowych: BAZA i BOMPOL.

Pakiet programowy BAZA, umożliwiający użytkownikowi uzyskanie gotowych programów obsługi bazy danych o konfiguracji określonej przez użytkownika, stanowi oryginalne opracowanie zespołu autorskiego systemu STEP.

Pakiet programowy BOMPOL stosowany do zakładania modyfikacji zbiorów bazy danych przy wykorzystaniu programów obsługi bazy danych (o konfiguracji określonej przez użytkownika) jest opracowaniem opartym na pewnych zasadach logicznych standardowego pakietu programowego IBM *Bill of Material Processor (Version 2)* — przeznaczonego dla komputerów serii IBM/360.

Funkcja wyszukiwania informacji (o żądanej przez użytkownika strukturze i przekrojach) w podstawowej bazie danych realizowana jest przez system STEP 74 dzięki zastosowaniu pakietu programowego RAPORT.

Powiązanie pakietów programowych w zestawie typowym STEP 74 obrazuje rys. 1.



Rys. 1. Schemat powiązania pakietów programowych w zestawie typowym STEP 74

UWAGA: we wszystkich rysunkach użyto symboli kart perforowanych dla oznaczenia niektórych zbiorów oraz programów. Jednak możliwe jest oczywiście we wszystkich tych wypadkach zastosowanie magnetycznego nośnika informacji: taśmy lub dysku.

Architektura zestawu typowego pakietów programowych STEP 74

Pakiet programowy BAZA umożliwia użytkownikowi uzyskanie gotowych programów zakładania i permanentnej aktualizacji podstawowej bazy danych o konfiguracji określonej przez użytkownika:

- liczbą zbiorów głównych
- powiązaniami adresowymi między rekordami zbiorów głównych
- wzorami rekordów zbiorów głównych tj. liczbą i treścią pól informacyjnych.

Pakiet programowy BAZA składa się z poniżej podanych elementów.

JOZ — kompilator języka opisu zbiorów. Język opisu zbiorów (JOZ) służy do określania przez użytkownika konfiguracji bazy danych. Jest to język o charakterze deklaratywnym operujący minimalną liczbą prostych instrukcji (aktualnie 7).
Użytkownik definiuje konfigurację swojej bazy danych w języku JOZ. Program źródłowy napisany w języku JOZ poddany jest działaniu kompilatora JOZ. Po kompilacji otrzymuje się:

- komplet makroinstrukcji definiujących zbiory wejściowe i wyjściowe bazy danych (MAKRO)
- TAPA — tablicę definiującą nazwę danych (stanowi ona zbiór wejściowy dla programu GENER, który produkuje gotowe programy zakładania i aktualizacji bazy danych)
- wydruk elementów dokumentacji technicznej określonej bazy danych
- wydruk kontrolny przebiegu kompilacji JOZ wraz z diagnostyką błędów.

SEZAM — systemowy zbiór programów podstawowych — realizujących funkcje ogólne zakładania i aktualizacji podstawowej bazy danych.
Zbiór SEZAM stanowi drugi zbiór wejściowy dla programu GENER, produkującego gotowe programy zakładania i aktualizacji bazy danych określonej przez użytkownika.

MAKRO — systemowy zbiór makroinstrukcji, obejmujący:

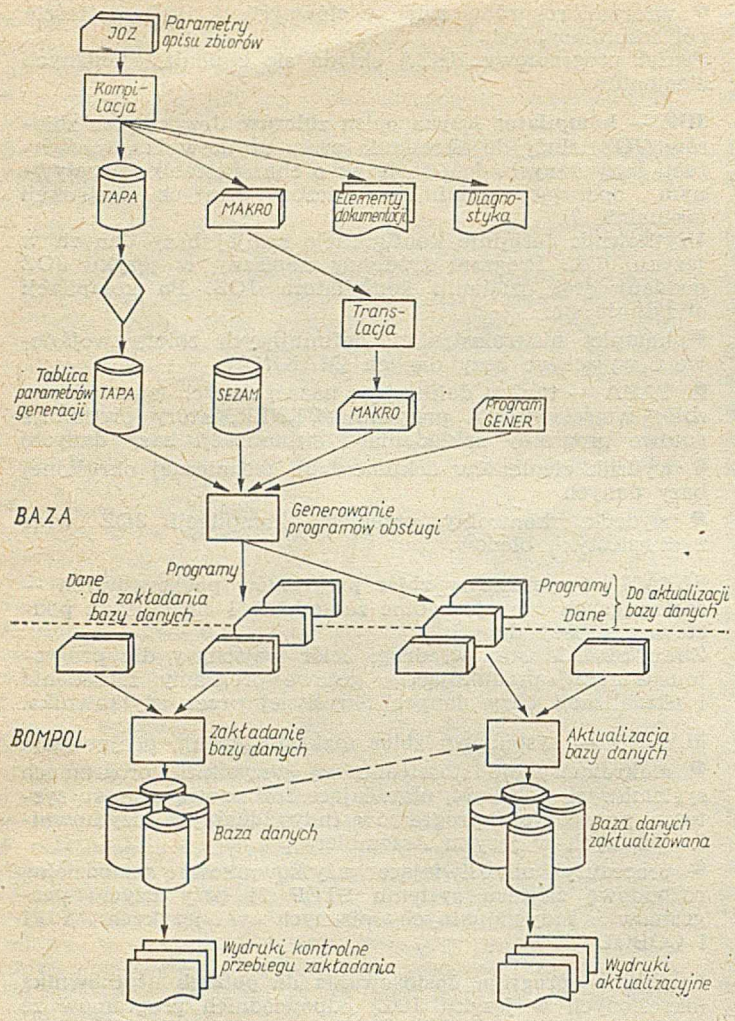
- makroinstrukcje stosowane we wszystkich programach systemowych STEP 74, ułatwiające rozbudowę zakresu systemu przy użyciu programów indywidualnych użytkownika pisanych w języku Assembler
- procedury umożliwiające użytkownikowi samodzielną rozbudowę zakresu systemu STEP 74 przy użyciu programów indywidualnych pisanych w językach: PL/1 i COBOL

GENER — program dostosowania do potrzeb użytkownika (określonych w języku JOZ) odpowiednich programów ze zbioru systemowego SEZAM w celu realizacji podstawowych funkcji pakietów programowych BAZA i BOMPOL.
Efektem działania programu GENER są gotowe programy (w formie modułów źródłowych) zakładania i aktualizacji podstawowej bazy danych o konfiguracji określonej przez użytkownika w języku JOZ.
Programy zakładania i aktualizacji podstawowej bazy danych wyprodukowane w wyniku działania pakietu programów BAZA używane są do fizycznego zakładania i fizycznej aktualizacji bazy danych dokonywanych przez pakiet programowy BOMPOL.
Wyprodukowane przez pakiet programowy BAZA programy obsługi podstawowej bazy danych mają postać modułów źródłowych w języku Assembler.
Przed przystąpieniem do fizycznego zakładania i fizycznej obsługi podstawowej bazy danych przy pomocy programów pakietu BOMPOL — programy obsługi podstawowej bazy danych winny być przetłumaczone i skatalogowane w systemowej lub prywatnej bibliotece modułów wynikowych (RL).
Tak przygotowane programy wraz z danymi użytkownika do zakładania bazy danych stanowią wejście do przebiegu zakładania tej bazy o konfiguracji określonej przez użytkownika.

Z kolei programy do aktualizacji bazy danych oraz dane aktualizacyjne przygotowane przez użytkownika stanowią wejście do przebiegu aktualizacji bazy danych.
Oba typy przebiegów realizowane przez pakiet programowy BOMPOL umożliwiają uzyskanie szeregu wydruków kontrolnych z zakładanej, bądź aktualizowanej bazy danych. Schemat działania pakietów programowych BAZA i BOMPOL obrazuje rys. 2.

Pakiet programowy RAPORT zastosowano w systemie STEP 74 w celu umożliwienia użytkownikowi wykonywania wydruków dotyczących wyszukanych informacji o dowolnym przekroju informacyjnym i poziomach sumowań — przy zastosowaniu języka JOW (Język Opisu Wydruku).
Pakiet obejmuje następujące elementy:

JOW — kompilator języka opisu wydruków. Język Opisu Wydruku (JOW) służy do określania przez użytkownika szaty graficznej, struktury informacyjnej i poziomów su-



Rys. 2. Schemat działania pakietów programowych BAZA i BOMPOL

mowań wydruku. Użytkownik poprzez podanie niewielkiej liczby parametrów określających treść informacyjną wydruku (nagłówki, wiersze szczegółowe i wiersze sumowań — do 5 poziomów) o szerokości wiersza 120—160 znaków otrzymuje gotowy program realizujący wydruk.

SELEKT — standardowy zbiór procedur selekcji odpowiednich informacji z bazy danych — realizujący funkcję wyszukiwania informacji określonych przez użytkownika. Poza opisanymi trzema podstawowymi pakietami programowymi w zestawie typowym STEP 74 zastosowano parametryzowany program WINO (Wyszukiwanie Informacji Określonej przez użytkownika) wzorowany w pewnej mierze na logice standardowego pakietu programowego IB/360 *Product Structure Retrioeval System.*

Program WINO umożliwia uzyskiwanie z bazy danych następujących informacji:

- rozwinięcie montażowe wyrobów złożonych wg pojedynczego poziomu złożenia czyli jednego stopnia montażu (ROZWINIĘCIE JEDNEGO POZIOMU)
- rozwinięcie montażowe wyrobów złożonych poziomami, aż do najniższego poziomu montażowego (ROZWINIĘCIE WEDŁUG POZIOMÓW)
- określenie ogólnej ilości każdego składnika, stanowiącego elementy składowe danego zespołu (ROZWINIĘCIE SUMARYCZNE)
- specyfikacja zespołów lub podzespołów, w których zastosowany jest bezpośrednio dany symbol części (ROZWINIĘCIE JEDNEGO POZIOMU)
- specyfikacja bezpośrednich i pośrednich elementów zastosowań danego symbolu części na wszystkich poziomach zastosowań (ZWINIĘCIE WEDŁUG POZIOMÓW)
- suma wszystkich bezpośrednich i pośrednich zastosowań danego symbolu części (ZWINIĘCIE SUMARYCZNE).

Metoda i kierunki rozbudowy systemu

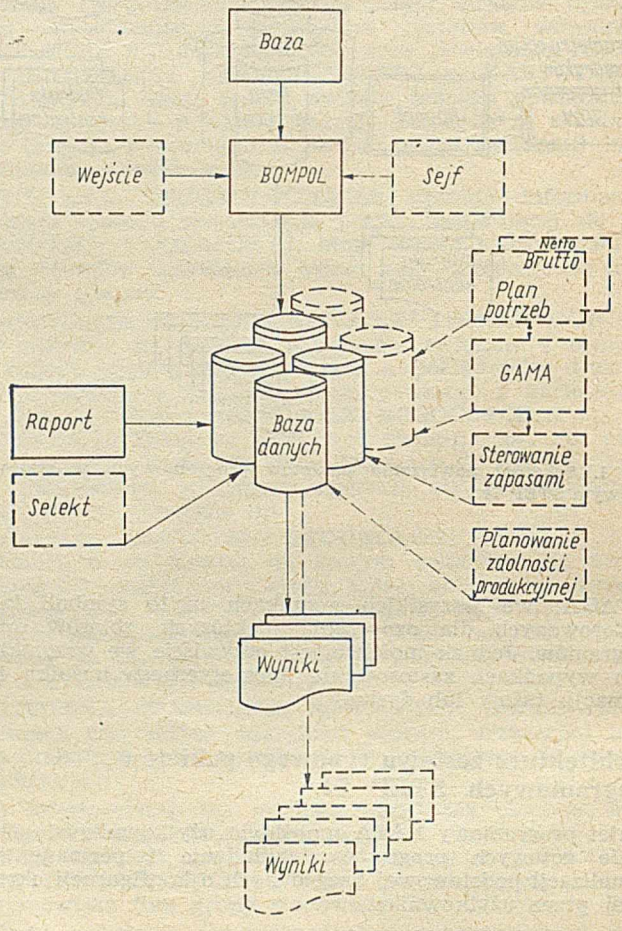
System STEP został opracowany w taki sposób, aby dzięki swej modułowej budowie tj. zestawowi odrębnych pakietów programowych wiązanych ze sobą za pomocą standardowych łączników — umożliwić użytkownikowi wybór odpowiedniego dla jego potrzeb zestawu modułów, a jednocześnie zapewnić użytkownikowi systematyczny rozwój funkcji i możliwości systemu dwiema podstawowymi drogami:

- rozbudowy systemu siłami własnymi użytkownika. W tym celu każdy pakiet programowy ma budowę otwartą i może być rozszerzony o nowe programy pisane przez użytkownika w językach programowania ASSEMBLER, PL/I i COBOL (w przyszłości możliwe będzie również programowanie w RPG).
- dołączania do posiadanej wersji systemu nowo opracowanych przez producenta pakietów programowych.

Kierunki rozbudowy zestawu typowego STEP 74 zobrazowano schematycznie na rys. 3.

W najbliższym okresie zestaw typowy zostanie wzbogacony o oryginalne pakiety programowe:

- WEJŚCIE — umożliwiający zastosowanie niestandardowego wprowadzania danych użytkownika do zakładania i aktualizacji bazy danych z dowolnego urządzenia WE/WY i zapewniający dokonywanie w bardzo szerokim zakresie kontroli formalnej i logicznej poprawności tych danych. Pakiet programowy WEJŚCIE umożliwi również zakładanie bazy danych systemu STEP bezpośrednio z innej bazy danych eksploatowanej dotychczas przez użytkownika.
- SEJF — umożliwiający automatyczne zabezpieczenie bazy danych systemu STEP oraz dokonywanie automatycznej reorganizacji tej bazy w określonym przez użytkownika zakresie.



Rys. 3. Kierunki rozbudowy systemu STEP

● SELEKT — pakiet programowy będący modernizacją i poważną rozbudową funkcji dotychczasowego zbioru procedur selekcji wyszukujących określone przez użytkownika informacje w podstawowej bazie danych oraz zbiorach pomocniczych.

W następnym etapie zestaw typowy zostanie rozbudowany o szereg pakietów programowych realizujących wydruki użytkowe z podstawowej bazy danych znacznie rozbudowanej w stosunku do obecnej postaci. Prace nad konstrukcją tych pakietów zostały już rozpoczęte i aktualnie koncentrują się na dwóch dziedzinach tematycznych:

— sterowanie zapasami w przedsiębiorstwie przemysłowym (pakiet GAMA i pakiet o roboczej nazwie STEROWANIE ZAPASAMI). Wydruki realizowane przez system dotyczyć będą zarówno gospodarki materiałowej jak i obrotu towarowego w przedsiębiorstwie

— planowanie produkcji oraz obliczanie zdolnień produkcji przedsiębiorstwa (pozostałe pakiety o nazwach roboczych uwidoczniono na rys. 3).

Funkcją celu oprogramowania przemysłowego jest dostarczenie kierownictwu przedsiębiorstwa przemysłowego systemu zarządzania, realizowanego za pomocą komputera umożliwiającego prawidłowe wykonywanie zadań przedsiębiorstwa przy najmniejszych nakładach czasu i kosztów.

Osiągnięcie tego celu wymaga racjonalizacji metod opracowywania oprogramowania przemysłowego, typizacji tego oprogramowania, tj. opracowywania wyspecjalizowanych w zakresie funkcji pakietów programowych powielarnych oraz racjonalizacji metod wdrażania systemów informatycznych w przemyśle.

Metoda produkcji oprogramowania przemysłowego przy użyciu nielicznych specjalistycznych grup projektowo-programowych umożliwia przede wszystkim:
- osiągnięcie wysokiej użyteczności produktu
- dużą liczbę zastosowań pojedynczych rozwiązań, co przy odpowiedniej kalkulacji kosztów zapewnia ekonomiczną efektywność produkcji
- specjalizację i stałe doskonalenie kadry
- eliminację szeregu wad klasycznych rozwiązań w zakresie systemów przemysłowych
- znaczne skrócenie cyklu projektowania.

Cykl projektowania klasycznych systemów przemysłowych był wieloetapowy (analiza, założenia, projekt techniczny, projekt techniczno-roboczy), po czym zwykle następował długi okres wdrażania systemu u użytkownika. Zwykle już w czasie eksploatacji próbnej pojawiała się konieczność modyfikacji systemu, co prowadziło do zmian programowych a nawet zmian założeń (gdyż np. w międzyczasie nastąpiły zmiany metod zarządzania). W procesie produkcji oprogramowania systemu STEP nie wydzielamy etapów cyklu projektowania. Praktycznie dokonywana jest tu permanentna analiza nowych rozwiązań w zakresie metod zarządzania oraz stale na bieżąco aktualizowany projekt techniczno-roboczy systemu. Ciężar odpowiedzialności za prawidłowość wdrożenia systemu przesunięty został znacznie w kierunku użytkownika. Wymaga to oczywiście znacznie wyższego stopnia organizacyjnego przygotowania użytkownika do wdrożenia systemu informatycznego niż w dotychczasowej praktyce, ale umożliwia zastosowanie u niego efektywnego i elastycznego systemu informatycznego znacznie mniejszym nakładem kosztów i w szeregu wypadków — w znacznie krótszym okresie czasu.

Podstawową korzyścią dla użytkownika, płynącą z zastosowanej metody produkcji typowych pakietów programowych jest możliwość posiadania systemu informatycznego o określonych przez niego funkcjach użytkowych, zmienianych w czasie w sposób elastyczny i o maksymalnie uproszczonej obsłudze.

Zwrotu „możliwość posiadania" w miejscu zwrotu „możliwość użytkowania" użyto tutaj świadomie. Użytkownik może zakupić i zainstalować zestaw pakietów programowych w dowolnym czasie, ale nie oznacza to dla niego konieczności natychmiastowego rozpoczęcia eksploatacji zestawu. Etapy: przygotowania organizacyjnego użytkownika oraz instalacji zestawu pakietów mogą być zrealizowane również w nietypowej kolejności odwrotnej.

Systematyczna rozbudowa zakresu systemu jest procesem możliwym do dowolnego rozłożenia w czasie. Użytkownik może konstruować swoją wersję systemu informatycznego w dowolnie długich odstępach czasowych, niezbędnych na opanowanie posiadanego już zakresu systemu oraz na dokonanie zmiany i usprawnień organizacyjnych przed kolejnym rozszerzeniem zakresu systemu, zarówno metodą dokupienia nowych elementów składowych, jak i metodą rozbudowy systemu własnymi siłami. Zapewniona jest również możliwość wymiany poszczególnych pakietów programowych na ich unowocześnione wersje, jak i możliwość zakupienia nowej wersji całego zestawu.

Ze względu na znaczny stopień uproszczenia obsługi systemu użytkownik nie musi posiadać własnej wysoko kwalifikowanej kadry dla eksploatacji systemu.

Wreszcie dzięki możliwości zastosowania wielu pojedynczych rozwiązań — koszt systemu jest kilkakrotnie niższy niż w rozwiązaniach klasycznych.

Produkcja oprogramowania przemysłowego w formie wyspecjalizowanych w zakresie funkcji pakietów programowych, możliwych do łączenia w różnorodne zestawy i wymiennych w razie potrzeby na rozwiązania nowsze umożliwia zasadnicze zmniejszenie dysproporcji między tempem zmian sytuacji gospodarczej (i związanych ściśle z nim procesem doskonalenia metod zarządzania), a tempem rozwoju oprogramowania przemysłowego.

Typizacja procesów programowania i utrzymywania systemów przemysłowych umożliwia konstrukcję i obsługę systemów informatycznych elastycznych, wolnych w dużej mierze od wad rozwiązań klasycznych. Grupa projektowo-programowa, wyspecjalizowana tematycznie zdolna jest do produkcji pakietów programowych o przeznaczeniu uniwersalnym, tj. możliwych do zastosowania u użytkowników o różnych strukturach organizacyjnych. Ścisła współpraca grupy projektowo-programowej ze specjalistami użytkownika pozwala na stałą konfrontację potrzeby użytkowników z możliwościami ich zaspokojenia przez projektowany system oraz eliminuje podstawową wadę rozwiązań klasycznych: utrwalanie sztywnych, zrutynizowanych algorytmów ewidencyjno-statystycznych wraz z aktualną strukturą organizacyjną. Wydaje się, że zastosowana przy budowie systemu STEP metoda jest krokiem naprzód w kierunku stworzenia metodyki projektowania elastycznego oprogramowania przemysłowego.

LITERATURA

PP-1-74-S-01/0 STEP 74. Opis systemu, OBRI Warszawa 1975
PP-1-74-S-06/0 STEP 74 Podręcznik systemowy, OBRI W-wa 1975 cz. I
PP-1-74-S-07/0 STEP 74. Podręcznik systemowy, OBRI W-wa 1975 cz. II

SYSTEMY OBIEKTOWE W ZPLiS

Regionalny Ośrodek Elektronicznych Maszyn Cyfrowych WSK PZL w Kaliszu zorganizował 25 listopada ub. roku konferencję resortową, poświęconą wykorzystaniu obiektowych systemów informatycznych dla usprawnienia organizacji i zarządzania przedsiębiorstwem. Ta problematyka była dyskutowana przez przedstawicieli ośrodków ETO Zjednoczenie Przemysłu Lotniczego i Silnikowego, w oparciu o założoną (i częściowo zrealizowaną) koncepcję systemu zarządzania dla WSK Kalisz. Przypomnijmy, że koncepcja ta oparta jest o system STEP-74, wdrażany w Kaliszu w oparciu o komputer JS — R-20. Zasadę bazowego systemu STEP-74 referował mgr inż. M. Klimek z OBRI, o konsekwencjach płynących z przejścia na automatyczną analizę wyników działalności gospodarczej przedsiębiorstwa — mówił mgr Z. Aulich z WSK Kalisz, natomiast o procesie budowy systemu obiektowego — kierownik ROEMC Kalisz, mgr Andrzej Staniszczak. Współorganizatorem tej pożytecznej konferencji było Koło Zakładowego PTE. (kb)

STANISŁAW MATWIN
Instytut Informatyki
Uniwersytetu Warszawskiego
MAREK MISSALA
Centrum Obliczeniowe
Polskiej Akademii Nauk

Czym są i co dają profile programów

W artykule omówiono profile statystyczne i dynamiczne programów, typowe metody ich otrzymywania oraz wyniki badań profilu dynamicznego kompilatora języka PASCAL.

Powstanie dużych i szybkich maszyn cyfrowych spowodowało, że programista stracił bezpośredni kontakt z maszyną. Jedyną otrzymywaną przez niego informacją są wydruki kompilatora: tekst programu, jego długość, czas wykonania. Daje się we znaki brak takich informacji, które zapewniłyby sprzężenie zwrotne między programistą, a efektem jego pracy — programem. Jednym z intensywnie rozwijanych ostatnio narzędzi programowych są tzw. profile statyczne i dynamiczne programów. Ich użycie dostarcza programiście informacji o zachowaniu jego programu w trakcie wykonania, a twórcom języków programowania i kompilatorów informacje o najczęściej używanych konstrukcjach języka. Ludziom zajmującym się nauczaniem programowania narzędzia te dają wiadomości o stylu programowania w danej grupie, o najczęściej popełnianych błędach.

Profil statyczny jest to informacja o liczbie wystąpień jednostek syntaktycznych w programie, stopniu kompilacji wyrażeń, głębokości i wielkości pętli.

Dość zaskakujące rezultaty otrzymane zostały w wyniku analizy dużej próby programów w FORTRANIE (razem ok. 250 000 kart) [1]. Analiza statyczna wykazała, że:
— prawie połowa instrukcji to podstawienia, są one zadziwiająco prostej postaci, połowa jest typu A=B
— pętle DO obejmują w większości 1—2 instrukcje
— mniej niż 10% pętli ma zagłębienie większe od 3
— 40% dodawań jest postaci $\alpha + 1$.

Profil statyczny programu rzutuje na czas jego kompilacji, lecz nie ma większego związku z czasem wykonania programu. Natomiast profil dynamiczny programu określa względny czas wykonania jego poszczególnych części. Możliwe są dwie strategie przy otrzymywaniu profilu dynamicznego programu.

● **Metoda liczników częstości wykonywania danych fragmentów programu.** Liczniki wstawiane są w odpowiednie miejsca w programie, aby można było np. obliczyć, ile razy wykonywała się każda instrukcja programu.
● **Metoda badania stanu programu.** W czasie wykonywania badanego programu procedura systemowa zapamiętuje w regularnych odstępach czasu wartość licznika rozkazów. Metoda ta daje dobre wyniki w przypadku, gdy jedynie nieduży procent czasu wykonania przetłumaczonego programu przypada na biblioteczne programy kompilatora. Odwrotna sytuacja ma miejsce w programach tłumaczonych z PL/1, gdzie znakomitą część kodu wynikowego stanowią odwołania do procedur bibliotecznych.
Profil dynamiczny programu pozwala stwierdzić, że większą część czasu liczenia pochłania nieduża pętla. Analiza i optymalizacja fragmentu programu może zmniejszyć czas wykonania nawet kilkakrotnie [2, 3]. Profil dynamiczny ma także zastosowania przy usuwaniu błędów logicznych. Pozwala on wydzielić te części programu, które nie były wykonywane (dla konkretnych danych). Oprócz tego profil dynamiczny może dostarczać informacji dla kompilatora optymalizującego, wskazując te fragmenty programu, które szczególnie opłaca się optymalizować.

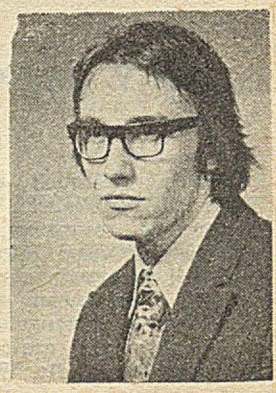
Opis typowych metod otrzymywania profili

Profil statyczny

Powszechnie używanym językiem w średnich i dużych instalacjach przemysłowych jest COBOL. Ambitny projekt systemu do badania programów pisanych w tym języku realizowany jest na Uniwersytecie w Guelph [4]. Inne dotychczas przeprowadzane próby badania profili programów dotyczyły w głównej mierze środowisk uniwersyteckich, istotnie różniących się od środowisk przemysłowych. System pisany jest w języku PL/1 z tym, że po przezwyciężeniu trudności związanych z programowaniem, system ma być przetransponowany na COBOL tak, by mógł być używany we wszystkich instalacjach.
Aktualnie system może dostarczyć profil statyczny dla jednego przebiegu programu oraz historię wszystkich przebiegów danego programu wraz z informacją o częstotliwości wykonywania.

Mgr Stanisław MATWIN ukończył Wydział Matematyki i Mechaniki Uniwersytetu Warszawskiego, Sekcja Metod Numerycznych (1972). Obecnie jest doktorantem Instytutu Matematycznego PAN. Zajmuje się językami programowania i metodami translacji. Uczestniczy w pracach nad kompilatorem języka PASCAL dla maszyn IBM serii 360/370 w pracowni Metod Programowania Centrum Obliczeniowego PAN.

Mgr Marek MISSALA jest absolwentem Wydziału Matematyki i Mechaniki Sekcji Metod Numerycznych Uniwersytetu Warszawskiego (1973). Pracuje w Centrum Obliczeniowym PAN i zajmuje się językami programowania i ich kompilatorami, a szczególnie językiem PASCAL.



Przy pierwszym przebiegu programu tworzony jest plik, do którego dopisywane są przy każdym następnym przebiegu nowe rekordy zawierające datę i czas liczenia, nazwisko autora i nazwę programu, częstość występowania różnych konstrukcji COBOLu oraz diagnostykę kompilatora. Rekordy te przygotowywane są przez jeden program systemu o nazwie EDIT tak, że zmiana kompilatora COBOLu wymaga tylko pewnych zmian w programie EDIT.

Z systemu można korzystać za pomocą kilku procedur, które umożliwiają:

— zapamiętanie statystyki dla danego przebiegu

— zapamiętanie i wydrukowanie statystyki dla danego przebiegu

— zapamiętanie statystyki dla danego przebiegu i wydrukowanie całej historii wszystkich przebiegów danego programu.

Użytkownik systemu ma kontrolę nad plikami, które zawierają historię jego programów i może je w dowolnym momencie skasować.

Wyniki otrzymane z systemu obejmują:

— listę konstrukcji używanych w działach: Identyfikacji Zestawu Maszynowego, Danych, z zaznaczeniem konstrukcji nie należących do standardu ANSI

— listę czasowników z działu procedur i ich częstości występowania

— inne wielkości, np. liczbę rekordów wejściowych, liczbę instrukcji w dziale procedur itp.

— tabelkę obrazującą występowanie i rodzaj błędów wykrytych przez kompilator.

Opisany powyżej system jest gotowy do eksploatacji. Czas pracy systemu wynosi ok. 70% czasu kompilacji i wiązania (*link*) programów analizowanych przez system. Trwają prace nad rozbudową systemu, projektuje się dołączenie możliwości badania dynamicznego zachowania się programów.

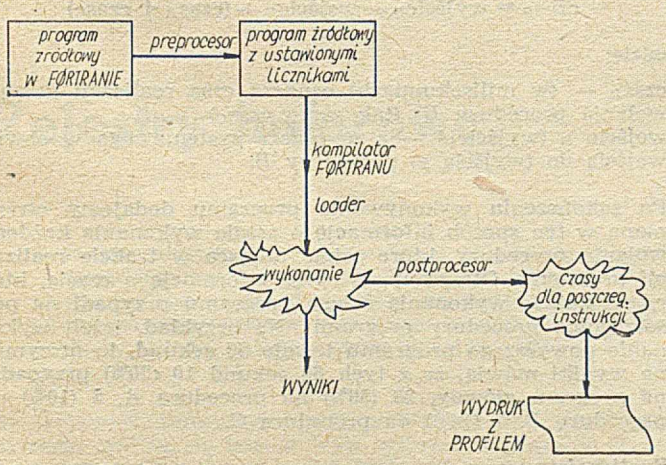
Profile dynamiczne

W czasie przeprowadzonych na uniwersytecie Stanford badań nad charakterystyką typowego programu w FORTRANIE, pojawił się problem obserwacji dynamicznego zachowania programów, a szczególnie lokalizacji najczęściej wykonywanych fragmentów programu.

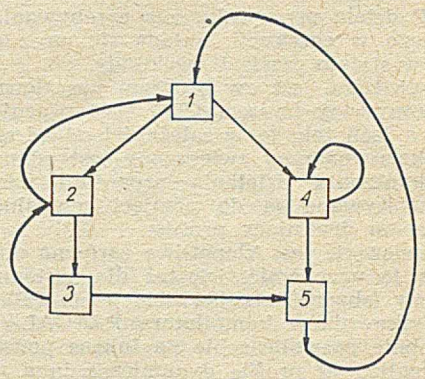
Na początku wydawało się, że najłatwiej będzie rozwiązać ten problem przez okresowe badanie wartości licznika rozkazów w trakcie wykonywania programu. Jednakże uzyskana w ten sposób informacja była raczej nieczytelna, można się było np. dowiedzieć, że duży procent czasu wykonania programu przypadł na instrukcje o adresach absolutnych 015928—015A28. Następnie po konsultacji z mapą *loadera* okazuje się, że są to instrukcje z wnętrza bibliotecznego podprogramu kompilatora. Jak widać metoda ta nie dawała pożądanego efektu, zwłaszcza gdy wyniki stanowić miały przedmiot analizy przeciętnego programisty [1]. Wtedy właśnie zbudowany został przez jednego z uczestników stanfordzkiego seminarium system FORTUNE [2], który uzupełnia wydruk postaci źródłowej programu informacją o liczbie i łącznym czasie wykonań każdej instrukcji. Informacja ta umieszczona jest przy każdej instrukcji programu, co umożliwia stosunkowo łatwą analizę wyników. Metoda zastosowana w systemie FORTUNE jest bardzo prosta — za każdą instrukcją wstawiony zostaje licznik, który za każdym przejściem sterowania przez tę instrukcję zwiększany jest o 1. Po wykonaniu programu wartości liczników oraz ich iloczyny przez szacunkowe czasy wykonania typowych instrukcji dołączone są do zapisanego na dysku programu źródłowego.

System kończy swą pracę drukując, poza normalnymi wynikami programu, informacje o profilu czasu wykonania [2]. Schematyczne działanie systemu przedstawiono na rys. 1. Metoda wstawiania liczników, jest rzecz jasna, tym lepsza, im mniej liczników trzeba wstawić. Osiągnięto tu eleganckie wyniki teoretyczne 5,6, które warto przedstawić w kilku zdaniach.

Cały problem polega na mierzeniu przepływów wierzchołkowych w grafie (rys. 2), będącym w tym przypadku schematem blokowym programu.



Rys. 1.



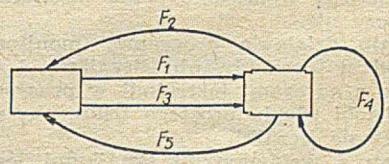
Rys. 2.

Krawędź od 5 do 1 (ozn. F_{51}) dodana jest tylko po to, aby zachowany był przepływ w całym grafie (tj. dla każdego wierzchołka przepływ wejściowy równy jest przepływowi wyjściowemu). Można wtedy zastosować tzw. prawo Kirchhoff'a dla grafów mówiące, iż przepływ w całym grafie wyznaczany jest przez przepływ poza drzewem rozpinającym graf (na rys. grube kreski). W ogólnym przypadku w grafie *om* krawędziach i *n* wierzchołkach wystarczy zmierzyć $m-n+1$ przepływów krawędziowych (w naszym przykładzie będzie to $9-5+1=5$ przepływów). Wszystkie pozostałe wyznaczyć można w oczywisty sposób, np.:

$$F_{23} = F_{32} + F_{35}$$

$$F_{12} = F_{23} + F_{21} \quad \text{itd.}$$

Rzeczywistym przedmiotem zainteresowania są jednak nie przepływy krawędziowe, a przepływy wierzchołkowe. Istnieje algorytm opublikowany po raz pierwszy w pracy [5] i pozwalający przekształcić każdy graf *A* o zachowanym przepływie w taki graf *B*, którego przepływy krawędziowe równe są przepływom wierzchołkowym *A*. Np. w wyniku stosowania algorytmu do powyższego grafu otrzymamy graf przedstawiony na rys. 3. Jeżeli z kolei do tego grafu



Rys. 3.

zastosować prawo Kirchhoff'a to okaże się, że trzeba zmierzyć tylko $5-2+1=4$ przepływy krawędziowe, przy czym musi być mierzone F_4. Każdy z otrzymanych stąd przepły-

[1] Dokładna analiza opisanej sytuacji wykazała, że są to procedury wejścia-wyjścia. Ponieważ przeciętnie zajmują one 2% tekstu programu, a 25% czasu wykonania, opłacalnym wydaje się zamiana przynajmniej niektórych z nich na podprogramy otwarte (tzw. *in-line I/O editing*).

[2] System ten został ostatnio zrealizowany w Instytucie Informatyki UW.

wów gałęziowych jest poszukiwanym przepływem wierzchołkowym grafu *A*.
W pracy [6] pokazano, że algorytm jest optymalny, tj. zmierzenie mniejszej liczby przepływów nie wystarcza do wyznaczenia pozostałych. Metodę tę wykorzystano praktycznie [8], mierząc tylko niektóre przepływy krawędziowe spoza drzewa rozpinającego podgraf, a następnie automatycznie wyznaczając przepływy wierzchołkowe.

Inne przykłady realizacji profili dynamicznych

Poniżej przedstawione są sposoby budowy narzędzi do otrzymywania profili dynamicznych dla starego kompilatora PASCALA (z 1972 r.) [9, 10] zrealizowane dwoma, opisanymi na początku artykułu, metodami.
Autorzy kompilatora wykorzystali specyficzną cechę maszyn systemu CDC 6000. Mianowicie system CDC 6000 składa się z jednego lub dwóch dużych procesorów i kilkunastu małych, mających własną pamięć procesorów peryferyjnych (PP). Program o nazwie FQC *(FreQuency Counting)* składa się z programu dla PP oraz z programu, który na podstawie przygotowanych przez pierwszy program liczników drukuje profil dynamiczny programu przetłumaczonego z PASCALA.
Program PP działa przez cały czas przebywania programu pascalowskiego w egzekucji. Składa się on z pętli wykonującej się w stałym czasie 1 msekundy.
W pętli sprawdzane jest czy program pascalowski jest aktualnie wykonywany (system CDC 6000 pracuje z podziałem czasu), a jeśli tak, to w zależności od adresu wykonywanego rozkazu maszyny zwiększany jest odpowiedni licznik. Po zakończeniu działania programu pascalowskiego liczniki przepisywane są do pamięci centralnej. Zaczyna działać program drukujący profile.
Opisane rozwiązanie jest kłopotliwe zarówno dla tych, którzy próbują je uruchomić, jak też dla systemu operacyjnego maszyny (absorbuje całkowicie jedno PP).
Jeden z użytkowników kompilatora PASCALA nie mogąc prawdopodobnie pozwolić sobie na luksus powyższego rozwiązania zrealizował profile dynamiczne inną metodą [11]. Zmodyfikował mianowicie generator kodu kompilatora tak, by sam kompilator wstawiał liczniki i kod, który je zwiększa do kodu wynikowego. Jest to strategia o wiele bardziej efektywna od procesora wpisującego liczniki do programu źródłowego.
Kompilator języka PASCAL napisany jest w PASCALU. Składa się z rekursywnych procedur. Kod dla każdej instrukcji generowany jest przez procedurę STATEMENT, która np. woła procedurę COMPSTAT (COMPound STATement), gdy trzeba wygenerować kod dla instrukcji BEGIN... END. Dlatego wystarczyło zmodyfikować tylko jedną procedurę STATEMENT, aby kompilator generował liczniki dla wszystkich instrukcji programu. Dodatkowe ułatwienie wynikało z faktu, że kompilator nie przeprowadza optymalizacji globalnej i z jednej instrukcji do drugiej nie jest przekazywana w większości rejestrów żadna informacja.
Licznik wraz z rozkazami maszyny, które go zwiększają zajmuje dwa słowa pamięci (60 bitowe) i jest przechowywany wraz z kodem programu. Zwiększenie licznika trwa 1,2 μs. Możliwe są inne rozwiązania, np. licznik wraz z instrukcją skoku do podprogramu zajmuje jedno słowo pamięci lecz czas wykonania podprogramu wynosi 5 μs. Wybór pierwszego rozwiązania oszczędzającego czas kosztem pamięci oparty jest na założeniu, że wiekszość programów z zastosowań naukowych posiada dużo danych, a mało instrukcji.
W tak zmodyfikowanym kompilatorze języka PASCAL istnieje mechanizm, przy użyciu którego programista decyduje, czy mają być wstawiane przed instrukcjami programu liczniki, czy nie.
Po zakończeniu działania programu, monitor PASCALA przegląda kod w poszukiwaniu liczników i wypisuje ich wartości wraz z adresami instrukcji w programie. Możliwe jest nawet, dzięki użyciu prostego triku, aby sam program przetwarzał swoje liczniki.
Kod programu z wstawionymi licznikami jest o ok. 30% większy i wykonuje się 15% dłużej. Liczby te byłyby niewątpliwie lepsze, gdyby stosować prawo Kirchhoff'a w celu ominięcia zbędnych liczników.

Nasze doświadczenia z profilami

Przystępując do realizacji naszego systemu pomiaru czasu wykonania programów przyjęliśmy następujące założenia:
- będziemy badać programy napisane w PASCALU
- jesteśmy zainteresowani badaniem dużych programów, zawierających kilka tysięcy instrukcji
- system ma być na tyle prosty, aby realizacja nie wymagała dużego nakładu pracy związanej z jego zaprogramowaniem.

W konsekwencji zadecydowaliśmy, że system dostarczać będzie informacji o względnej ilości czasu wykonania przypadającego na każdą procedurę programu w stosunku do czasu wykonania całego programu. Jest to istotne ograniczenie w stosunku do założeń projektów opisanych powyżej, jednakże następujące argumenty wydają się uzasadniać to ograniczenie:
- procedura jest podstawową konstrukcją używaną przez programistę tworzącego program w PASCALU, zwłaszcza przy budowie dużego programu projektowanego techniką zstępującą *(top-down)*. Wszystkie znane nam kompilatory języka PASCAL celowo zawierają sztywne ograniczenia na długość procedury
- rozszerzenie systemu w taki sposób, aby brane były pod uwagę mniejsze fragmenty programu prowadziłoby do dużego nakładu pracy przy budowie systemu
- w rzeczywistych, dużych programach „szpiegowanie" przebiegu wykonania bardziej podstawowych konstrukcji języka dostarczałoby tak dużej ilości informacji, że w sumie stawałaby się ona mało czytelna. Weźmy np. system podający względne częstości wykonania jako ciągi gwiazdek (lub liczby) po prawej stronie wydruku każdej linii programu.

Przy analizie wyników dla dużego programu trzeba linia za linią przejrzeć kilkudziesięciostronicowy wydruk!
W naszym systemie po prostu odejmujemy od stanu zegara w chwili wyjścia z procedury jego stan w chwili wejścia do tej procedury.

Np. dla procedury B w następującym programie:

```
program P;
  procedure A;
    begin
      (* A nie zawiera wołań procedur *)
      ...
    end (* A *);
  procedure B;
    procedure C;
      begin
        (*C nie zawiera wołań procedur*)
        ...
      end (* C *)
    begin (* B *)
      ... A; ... C; ...
      (* w B wołane są tylko procedury A i C *)
    end (* B *);
  begin (* P *)
    ... B; ...
  end . (* P *)
```

otrzymujemy:

$$czas_B = wyjście_B - wejście_B - (czas_A + czas_C)$$

gdzie:

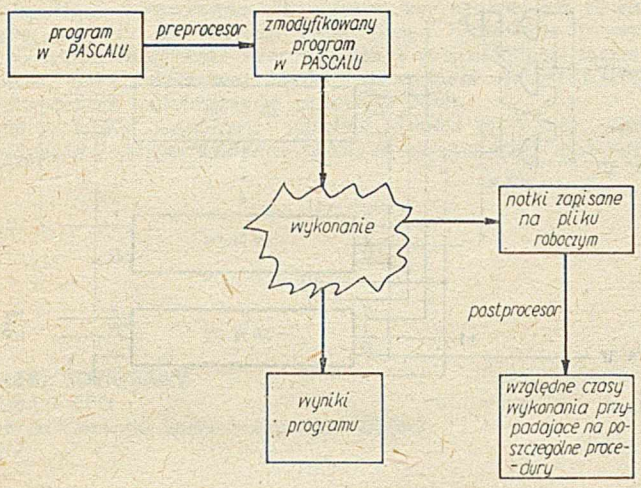
$czas_B$ — (w milisekundach) oznacza czas realizacji danego wołania procedury *B;* (tak samo $czas_C$)
$wejście_B$ ($wyjście_B$) — stan zegara systemowego w chwili wejścia do (wyjścia z) procedury *B*.

Po zakończeniu wykonywania programu dodajemy otrzymaną w ten sposób informację o czasie wykonania każdego wołania procedury, które miało miejsce w trakcie realizacji programu. Otrzymujemy więc informację o czasie, który w trakcie wykonania całego programu przypadł na poszczególne procedury programu. Na przykład, jeżeli wykonanie powyższego programu trwało 50 sekund, to otrzymane wyniki mówią, że z tych 50 sekund 10 (20%) przypadło na program główny, 28 (56%) na procedurę *A*, 5 (10%) na procedurę *B*, 7 (14%) na procedurę *C*.

Realizacja

System składa się z dwóch części, preprocesora i postprocesora. Preprocesor wstawia na początku i na końcu procedury instrukcje PASCALA powodujące przy wykonaniu procedury zapisanie na pliku roboczym informacji postaci:
(nazwa, czas, rodzaj punktu),
gdzie:
— nazwa jest nazwą procedury
— czas jest stanem zegara systemowego w danej chwili
— rodzaj punktu jest zmienną, której wartość określa czy

informacja odnosi się do wejścia, czy do wyjścia procedury.
Aby można było wstawić tego typu instrukcje, preprocesor musi rozpoznawać w tekście programu początek i koniec procedury. W trakcie wykonania programu instrukcje wstawione przez preprocesor powodują wygenerowanie notek opisanej wyżej postaci i zapisanie ich na pliku roboczym.



Rys. 4.

Po zakończeniu wykonania programu plik ten jest przetwarzany przez postprocesor, który z pliku roboczego czyta informacje postaci
wejście$_P$ wejście$_B$ wejście$_A$ wyjście$_A$ wejście$_C$ wyjście$_C$ wyjście$_B$ wyjście$_P$,
dopasowuje parami poszczególne notki, a następnie sumuje dla każdej procedury czasy przypadające realizacjom jej poszczególnych wołań.
Schematycznie pracę systemów przedstawiono na rys. 4.
Systemu naszego użyliśmy do zbadania profilu dynamicznego kompilatora PASCALA, który zapisany jest w języku PASCAL; danymi był tekst kompilatora. Otrzymane wyniki nie zawierają rewelacji, ale mówią dokładnie, ile czasu kompilacji pochłania analiza leksykalna, ile generacja kodu, optymalizacja użycia rejestrów itp.

Kilka uwag końcowych

Wydaje się nam, że profile zaczynają zdobywać trwałą pozycję w programowaniu. W przyszłości prawdopodobnie każdy system programowania zawierać będzie tego rodzaju narzędzia. Dzisiaj zaczynają one wchodzić np. w skład niektórych pakietów IBM-owskich, są także realizowane sprzętowo w minikomputerach.
Wykorzystanie profili zmienia w istotny sposób metodę pracy programisty i pozwala na stosunkowo łatwe otrzymywanie efektywnych programów, co daje oczywiście ich autorowi wiele satysfakcji.

LITERATURA

[1] Knuth D. E.: „An Empirical Study of FORTRAN Programs", SOFTWARE-PRACTICE AND EXPERIENCE, Vol. 1, 105-133 (1971).
[2] Ingalls D.: „Execution time profile as programming tool", DESIGN OPTIMIZATION OF COMPILERS, R. Rustin, Ed. Courant Comp. Sci. 5. NJ 72.
[3] Samet P. A.: „Detailed Analysis of a Program — an Instructive Horror Story", SOFTWARE-PRACTICE AND EXPERIENCE, Vol. 5, 211-213 (1975).
[4] Salvadori A., Gordon J., Capstick C.: „Static Profile of COBOL Programs", SIGPLAN NOTICES, Vol. 10, No 8, 20—33 (1975).
[5] Nahapetian A.: „Node Flows in Graphs with Conservative Flow", ACTA INFORMATICA, Vol. 3, 37—41 (1973).
[6] Kunth D. E., Stevenson F. R.: „Optimal Measurement Points for Program Frequency Counts", BIT, Vol. 13, 313—322 (1973).
[7] Kunth D. E.: „The Art of Computer Programming", Vol. 1 (1968), Addison-Wesley.
[8] Nahapetian A.: „Aids for the analysis of assembly language programs", Stanford University Engineer thesis (1972).
[9] Matwin St.: „Język programowania PASCAL i jego kompilacja", INFORMATYKA, 1974, nr 11.
[10] Missala M.: „Programowanie w języku PASCAL 6000", PRACE CO PAN, No 179 (1975).
[11] Yuval G.: „Gathering Run-Time Statistics Without Black Magic", SOFTWARE-PRACTICE AND EXPERIENCE, Vol. 5, 105—108 (1975).

HENRYK WIĘCEK
Instytut Maszyn Matematycznych
Oddział Śląski
Katowice

Rejestry buforowe

Podano klasyfikację i nomenklaturę pamięci buforowych. Omówiono pamięci buforowe zbudowane wyłącznie z rejestrów przesuwających (rejestry butorowe). Zaproponowano realizację rejestrów buforowych (FIFO i LIFO) w oparciu o układy scalone średniej integracji MSI.

W systemach informatycznych, w których dochodzi do nasycenia, tworzą się tzw. kolejki zgłoszeń. Zgłoszenia w kolejce uporządkowane są w kolejności ich przybywania. Sposoby obsługi kolejek określające kolejność wybierania zgłoszeń dzieli się na:

— pierwszy przybył, pierwszy obsłużony *(First-In, First-Out,* w skrócie FIFO).
Do obsługi pobiera się zgłoszenie o najdłuższym czasie oczekiwania

— ostatni przybył, pierwszy obsłużony *(Last-In, First-Out,* w skrócie LIFO).
Do obsługi pobiera się zgłoszenie o najkrótszym czasie oczekiwania

— obsłużony wg priorytetu.
Do obsługi wybiera się zgłoszenie, któremu przypisano najwyższy priorytet.

W przypadku sprzętowej *(hardwarowej)* obsługi kolejek, kolejki zgłoszeń przechowywane są w specjalnych typach pamięci buforowych. Pamięci buforowe wybierające zgłoszenia wg najkrótszego czasu oczekiwania nazywamy stosami, natomiast pamięci wybierające zgłoszenia wg najdłuższego czasu oczekiwania nazwiemy — ze względu na dotychczasowy brak określenia — silosami. Terminologię spotykaną w literaturze zachodniej podano w tabeli I.

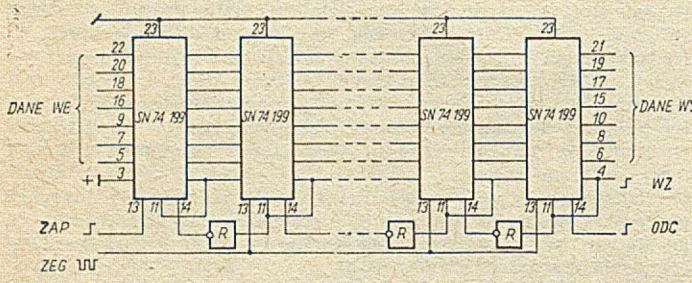
W dalszym ciągu omówione zostaną pamięci buforowe zbudowane wyłącznie z rejestrów przesuwających, czyli tzw. rejestry buforowe.

TABELA I. Nomenklatura anglosaska pamięci buforowych

Terminologia	Reżim pracy	Nazwa pamięci buforowej
amerykańska	FIFO LIFO	stack
niemiecka	Silobetrieb Stapelbetrieb	Siloregister Stapelregister

Rejestr buforowy typu FIFO (silos)

Logiczne urzeczywistnienie zasady First In — First Out przedstawione jest na rys. 1. Układ składa się z szeregu kaskadowo połączonych rejestrów przesuwających, pracujących w reżimie równoległym (LOAD). Wczytana informacja przesuwana jest automatycznie dalej i składowana na wyjściu silosa. Wpisywanie, przesuwanie i odczytywanie informacji zachodzi całkowicie niezależnie od siebie (asynchronicznie) pod wpływem impulsów zapisujących, taktujących przesuwanie i odczytujących.



Rys. 1.

Impulsy zapisujące podawane są na wejście 13 pierwszego rejestru przesuwającego. Strobują [1]) one zapis informacji podawanej równolegle na wejścia 3 ... 22. Pozostałe wejścia 13 połączone są ze źródłem zewnętrznych impulsów taktujących (zegarem). Wejścia 3, 11, 14 i wyjścia 4 poszczególnych rejestrów tworzą warstwę sterującą, kierującą przesuwaniem wczytanej informacji w takt impulsów zegarowych, aż do ostatniej wolnej pozycji bufora. Każda zajęta pozycja sygnalizowana jest „1" na wyjściu 4 *(marker)*, która uniemożliwia wpisanie nowej informacji na tę pozycję wskutek sprzężenia 4 z 11 i zablokowania impulsów przesuwających. Dopiero podanie impulsu odczytującego na wejście 14 ostatniego rejestru powoduje wyzerowanie zajętej pozycji, zgaszenie markera i odblokowanie impulsów taktujących, co w konsekwencji prowadzi do stopniowego przemieszczania informacji w prawo w rytm impulsów zegarowych.

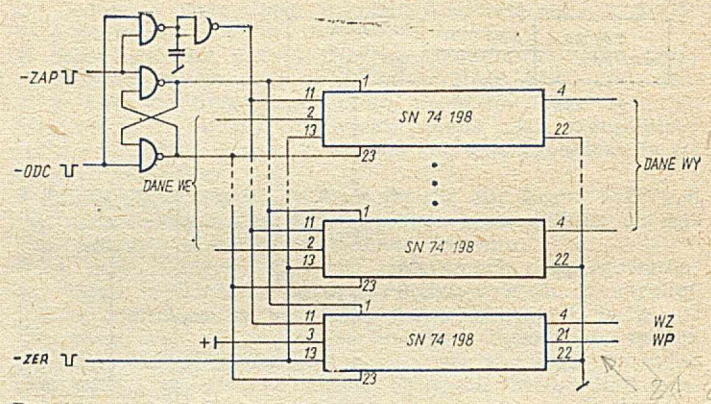
Układy różniczkujące, oznaczone na rysunku symbolem *R*, przeznaczone są do zerowania poszczególnych pozycji rejestru buforowego. Można je zrealizować z elementów dyskretnych *RC*, jak i przerzutników *D*. Sygnał ogólnego zerowania nie musi być podawany na wszystkie pozycje rejestru równocześnie. Wystarczy, że na czas jego trwania impulsy zegarowe sumowane będą z wejściem strobującym odczyt 14.

Rejestr buforowy typu LIFO (stos)

Układ bufora typu LIFO przedstawiony jest na rys. 2. Zbudowany jest z szeregu rejestrów przesuwających, pracujących w reżimie szeregowym (SHIFT) oraz układu asynchronizującego zapis i odczyt. Każdemu zapisowi nowej informacji towarzyszy przesunięcie starej zawartości w prawo, a odczytowi — przesunięcie w lewo. Ostatnio zapisana informacja odczytywana jest jako pierwsza. Równoczesny zapis i odczyt informacji nie jest możliwy.

[1]) Strobuje — taktuje, bramkuje wg Dictionary of Data Processing (english-german-french) Wyd. Elsevier Co. 1973 strobe pulsce = Taklimpuls = impulsion de rythme.

Impulsy zapisujące i odczytujące podawane są na wejścia przerzutnika asynchronicznego *RS* i układu opóźniającego. Przerzutnik ustawia kierunek przesuwania danych w rejestrach przesuwających, a układ opóźniający strobuje wczytywanie i odczytywanie danych.

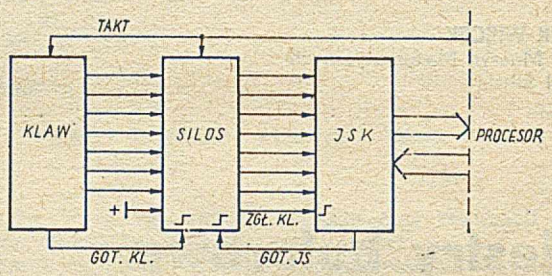


Rys. 2.

Ustawienie kierunku przesuwania w prawo równoznaczne jest z przygotowaniem układu do zapisywania, a ustawienie w lewo — do odczytywania. Informacja wejściowa wpisywana jest równolegle poprzez wejścia 2, a odczytywana poprzez wyjścia 4. Jeden z rejestrów przesuwających spełnia rolę wskaźnika zapełnienia *WZ* i przepełnienia *WP*. Wejście 2 tego rejestru podłączone jest do „1" logicznej.

Zastosowanie

Pamięci buforowe typu FIFO znalazły największe zastosowanie w przemysłowych układach sygnalizacji, rejestracji i sterowania, natomiast pamięci LIFO — prawie wyłącznie w nowoczesnych kalkulatorach i niektórych procesorach. Przykładowe zastosowanie silosa do buforowania danych z klawiatury alfanumerycznej pokazano na rys. 3. Każde



Rys. 3.

naciśnięcie klawisza powoduje wygenerowanie 7-bitowego kodu ASCII i opóźnionego sygnału gotowości klawiatury. Klawisz *GOT. KL* strobuje wczytanie kodu i „1" logicznej do pamięci buforowej. Po przemieszczeniu „1" na wyjście pamięci, generowany jest sygnał *ZGŁ. KL.* Sygnał obsługi zgłoszenia *GOT JS* powoduje pobranie kodu do jednostki sterującej klawiatury i wyzerowanie odczytywanego znaku wraz z sygnałem *ZGŁ. KL.* Impulsy zegarowe dla silosa i klawiatury skanningowej [2]) brane są ze wspólnego źródła. Ze względu na stosunkowo małą liczbę wprowadzeń kores-

[2]) Klawiatura skanningowa (scanning keyboard) — klawiatura działająca na zasadzie sekwencyjnego wybierania i porównywania kodu poszczególnych klawiszy ze stanem napędzanego licznika wg Słownika naukowo-technicznego (angielsko-polskiego) Wyd. WNT, Warszawa 1973 scanning = wybieranie

pondujących z otoczeniem, jak i możliwość uzyskania dużej liczby słów przy stałej liczbie wejść, pamięci buforowe świetnie nadają się do scalenia i upakowania w obudowie DIL. Dlatego też coraz więcej producentów oferuje układy buforów cyfrowych w postaci kostek scalonych LSI. Dla przykładu warto przytoczyć pamięci serii Macrologic 9400 firmy FAIRCHILD. A oto ich krótka charakterystyka:
9403 — FIFO, organizacja pamięci 16 słów po 4 bity, wprowadzanie i wyprowadzanie danych szeregowe lub równoległe, wejście od wyjścia całkowitego niezależne i asynchroniczne, możliwość szeregowego i równoległego łączenia kostek bez dodatkowej logiki, technologia Schottky TTL, maksymalna częstotliwość pracy 20 MHz.
9406 — LIFO, organizacja pamięci 16 słów po 4 bity, wprowadzanie i wyprowadzanie danych równoległe, możliwość zwiększania długości słowa przez równoległe połączenie kilku kostek bez dodatkowej logiki, technologia Schottky TTL, maksymalna częstotliwość 10 MHz.

LITERATURA

[1] Pragakis G.: Fast stacking — register buffer has minimum delays. ELECTRONIC ENGINEERING, 1972, April.

[2] Schaffernak A.: Schieberegister zur Steuerung industrieller Ordnungs prozesse. ELEKTRONIK, 1971, Heft 3.

[3] Digitaler Pufferspeicher (DSB) TMS 4024. TEXAS INSTRUMENTS Applikationsbericht EB 83.

[4] Praca zbiorowa pod redakcją Eimbindera: Pamięci półprzewodnikowe scalone. Warszawa, 1975. WNT, s. 129—132.

MAREK ZDZIUBANY
Ośrodek EPD
Fabryki Osprzętu Samochodowego POLMO
Łódź

Koncepcja modyfikacji standardowego systemu organizacji bazy danych „Bill of material processor" (BOMP)

Omówiono koncepcję przeprowadzenia modyfikacji standardowego systemu organizacji bazy danych BOMP firmy IBM przeznaczonego dla komputerów serii 360. Celem proponowanych modyfikacji jest lepsze dostosowanie systemu do potrzeb przedsiębiorstw przemysłowych poprzez uściślenie wyników, które można otrzymać operując informacjami zawartymi w zbiorach bazy danych.

BILL OF MATERIAL PROCESSOR jest uniwersalnym systemem organizacji bazy danych dla systemów informatycznych wchodzących w skład standardowego oprogramowania rodziny maszyn IBM 360. System BOMP może znaleźć zastosowanie w różnych dziedzinach działalności, jednak w przypadku zastosowania go do obsługi bazy danych przedsiębiorstwa przemysłowego wykazuje pewne braki. W systemie BOMP zbiory danych utrzymywane są na dyskach magnetycznych, co stwarza możliwość szybkiego dostępu do każdej informacji. Cztery zbiory danych działających w ramach tego systemu noszą nazwy:

- Kartoteka rodzajowa nazywana również kartoteką główną pozycji (ITEM MASTER FILE)
- Kartoteka struktury wyrobów (PRODUCT STRUCTURE FILE)
- Kartoteka technologiczna (STANDARD ROUTINE FILE)
- Kartoteka stanowisk roboczych (WORK CENTER FILE).

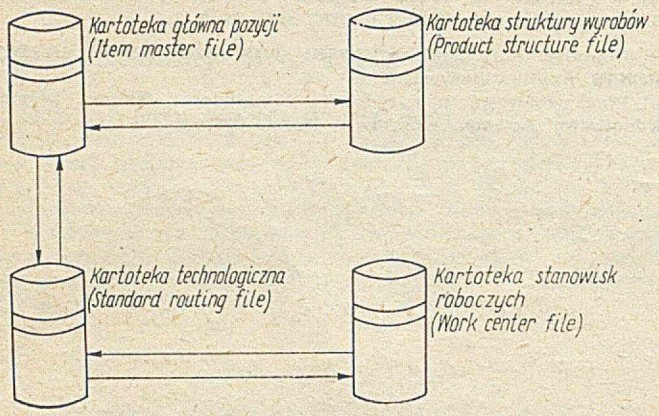
Kartoteka główna pozycji zawiera zespół informacji o produktach finalnych przedsiębiorstwa oraz ich składnikach prostych i złożonych. Kryterium kwalifikacyjnym jest bezpośredni udział elementu w więcej niż jednej podstawowej operacji technologicznej w danym przedsiębiorstwie. Tak więc np. jako element prosty traktowane będzie łożysko wmontowywane do silnika spalinowego, natomiast elementem złożonym będzie pierścień dystansujący użyty do montażu tego łożyska, jeżeli wycinany był on z blachy w wydziale pras danego przedsiębiorstwa i to nawet wtedy, jeżeli oprócz wycięcia nie wykonywano na nim żadnych innych operacji technologicznych. Identyfikatorem poszczególnych pozycji jest ich indeks, któremu system nie stawia praktycznie żadnych wymagań.
Omówione powyżej pojęcia wprowadzone zostały ze względu na nieprecyzyjność używanych powszechnie pojęć materiał i detal.

Kartoteka struktury wyrobów jest zbiorem informacji o wzajemnych powiązaniach pozycji umieszczonych w kartotece głównej.
Każdy element złożony mający zapis w kartotece głównej pozycji powinien mieć w kartotece struktury wyrobów odpowiadającą mu grupę zapisów (strukturę), określającą jednoznacznie jakie elementy i w jakich ilościach zostaną zużywane w procesie jego wytworzenia. System BOMP nie ogranicza liczby składników struktury, wymaga tylko ujednolicenia zasad uszeregowania składników w ramach wszystkich struktur zawartych w zbiorze. Zazwyczaj zasadą tą jest kolejność użycia w procesie produkcji, lecz może nią być także sekwencja indeksu.

Kartoteka technologiczna jest zbiorem danych o sposobie wytworzenia elementów złożonych zawartych w kartotece głównej. Zapisy tej kartoteki określić mają jednoznacznie, jakie operacje technologiczne, w jakich komórkach produkcyjnych, za pomocą jakich urządzeń oraz w jakiej kolejności powinny być wykonywane dla prawidłowego wytworzenia poszczególnych elementów złożonych. Określony jest w nich również czas trwania każdej operacji technologicznej.

Kartoteka stanowisk roboczych zawiera zazwyczaj dane o maszynach i urządzeniach zainstalowanych w komórkach produkcyjnych przedsiębiorstwa wraz z ich danymi eksploatacyjnymi.
System BOMP nie stawia wymagań dotyczących jednorodności budowy zapisów w poszczególnych zbiorach, z wyjątkiem jednolitego rozmieszczenia pól kluczowych i jednakowej długości zapisów w ramach zbioru. Nie jest również ograniczona wielkość zbioru.
Jedynym ograniczeniem jest pojemność pamięci dyskowych w użytkowanej konfiguracji komputera.

Zasadniczą różnicą między bazą danych zorganizowaną na taśmach magnetycznych a zbiorami systemu BOMP jest istnienie tzw. powiązań adresowych między zapisami zarówno tego samego, jak i dwu różnych zbiorów (patrz rysunek 1).



Rys. 1. Baza danych systemu BOMP — strzałki symbolizują łańcuchowe powiązania adresowe

Powiązania adresowe polegają na tym, że w zapisie jednego zbiorów umieszczony jest bezpośredni adres dyskowy logicznie związanej z nim informacji znajdującej się w innym zbiorze. Powiązania takie funkcjonują również pomiędzy zapisami tego samego zbioru.

Jako przykład podać można zapis elementu złożonego kartoteki głównej pozycji, w którym umieszczony jest adres zapisu pierwszego składnika tego elementu w kartotece struktury wyrobów.

Z kolei w ww. zapisie strukturalnym umieszczony jest adres zapisu znajdującego się w kartotece głównej pozycji zawierającego dane o tym składniku. Omówiony zapis strukturalny zawiera w sobie również adres do zapisu strukturalnego następnego składnika.

Kilka podobnego typu powiązań adresowych funkcjonuje między kartoteką główną pozycji a kartoteką technologiczną oraz między kartoteką technologiczną a kartoteką stanowisk roboczych.

Istnieje możliwość odszukania dowolnego zapisu kartoteki głównej pozycji i kartoteki stanowisk roboczych, a dzięki powiązaniom adresowym szybkiego dotarcia do logicznie z nimi związanych zapisów innych zbiorów.
Posługując się programami wykorzystującymi powiązania adresowe między danymi zawartymi w kartotece głównej pozycji i kartotece struktury wyrobów można, dokonując tzw. zwinięć i rozwinięć (plosions) ustalić:

— przy wytwarzaniu jakich elementów złożonych (części, podzespołów, zespołów, wyrobów) i w jakich ilościach używany jest określony element zarówno prosty (materiał), jak i złożony

— jaka liczba elementów prostych (materiałów) i złożonych (zespołów, podzespołów, części) niezbędna jest dla wytworzenia określonego elementu złożonego (wyrobu).

Oczywiście możliwości te istnieją nie tylko dla jednej sztuki elementu złożonego lecz również dla dowolnie ustalonej ich liczby, np. wynikającej z sumy zamówień złożonych przez odbiorców.

Posiadając te informacje można z kolei, wykorzystując dane zawarte w kartotece głównej pozycji i kartotece technologicznej oraz istniejące między nimi powiązania adresowe, obliczyć ogólny czas, w ciągu którego powinny pracować poszczególne stanowiska robocze, aby wyprodukować żądaną liczbę produktów, tzn. ustalić obciążenie tych stanowisk.

W praktyce planowania produkcji podstawową rzeczą, interesującą kierownictwo przedsiębiorstwa, jest stwierdzenie jaki jest stosunek obciążenia stanowisk do czasu, którym przedsiębiorstwo dysponuje na każdym z tych stanowisk. Możliwość taka istnieje poprzez wykorzystanie powiązań adresowych między kartoteką technologiczną a kartoteką stanowisk roboczych. Wykonanie tych obliczeń na komputerze jest bez porównania szybsze i nie obciążone błędami rachunkowymi, które występują przy wykonywaniu tej pracy sposobami tradycyjnymi. Wykonując obliczenia trzeba dla poszczególnych stanowisk określić dysponowany czas pracy. Przyjęcie metody statystyczno-szacunkowej, stosowanej przy wykonywaniu tych obliczeń sposobem tradycyjnym, polegającej na zastosowaniu współczynnika wykorzystania nominalnego czasu pracy, urządzeń produkcyjnych jednakowego dla wszystkich lub też zróżnicowanego w zależności od rodzaju urządzenia, nie może być uznane za właściwe.

Metoda ta dająca wyniki — niezbyt dokładne w przypadku obliczeń dotyczących dłuższych okresów czasu — zawodzi całkowicie w przypadku dokonywania obliczeń dla okresów krótkich. Przyczyną jest różny stopień zużycia poszczególnych egzemplarzy maszyn i urządzeń oraz różne terminy wykonywania czynności remontowych. Okazuje się, że maszyna, dla której stosując metodę statystyczno-szacunkową przyjęto, że przepracuje np. 85% czasu nominalnego w wyniku przekazania do remontu przepracuje tylko około 60% czasu. Oczywiście inny egzemplarz tego typu maszyny zainstalowany w innej komórce przepracuje pełne 100% czasu, lecz czasowe przemieszczenie tam produkcji dezorganizuje jej proces powodując również wzrost kosztów. Nieprawidłowości te można usunąć uwzględniając przy obliczaniu dysponowanego czasu pracy szczegółowy terminarz przeprowadzania czynności remontowych.

Aby umożliwić wykonanie tych obliczeń z wykorzystaniem informacji zawartych w bazie danych zorganizowanej w oparciu o system BOMP, należy przy projektowaniu kartoteki stanowisk roboczych przyjąć założenia nieco odmienne od przyjmowanych zazwyczaj.

Odmienność ich polega na tym, że projektując zapisy zbioru powinno się przewidzieć zapisy dla wszystkich elementów struktury organizacyjnej procesu produkcji od przedsiębiorstwa poczynając poprzez:

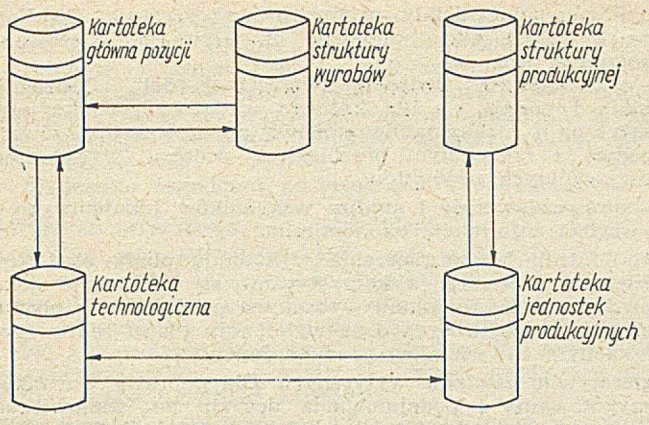
— zakłady
— wydziały
— oddziały
— gniazda
— linie
— grupy maszyn zamiennych
— egzemplarze maszyn i urządzeń.

Nazwy użyte powyżej dla określenia poszczególnych jednostek nie wymagają omówienia, gdyż przyjęte dla nich znaczenie zgodne jest z powszechną praktyką. Wyjątek stanowi termin „grupa maszyn zamiennych". Zazwyczaj za kryterium pozwalające zakwalifikować maszynę do grupy jest kryterium czysto techniczne, tzn. możliwość wykonania na wszystkich egzemplarzach zakwalifikowanych do grupy określonych operacji technologicznych. Niezbędne jest jednak przyjęcie, obok wyżej wymienionego, dodatkowego kryterium, które można nazwać organizacyjnym. W wyniku przyjęcia tego kryterium pod nazwą „grupa maszyn zamiennych" należy rozumieć jedno lub więcej stanowisk pracy wyznaczonych do wykonania określonych operacji technologicznych, tak rozmieszczonych, że skierowanie przypisanych im operacji na którekolwiek z nich nie spowoduje żadnych istotnych zmian w długości cyklu produkcyjnego, czasie wykonania, jakości oraz koszcie wykonania operacji.

Dla podkreślenia różnicy w logice wewnętrznej zbioru proponuje się używanie w stosunku do niego nazwy „kartoteka jednostek produkcyjnych" (rys. 2). Ze względu na szeroki zakres pojęcia „jednostka produkcyjna" kartoteka winna zawierać trzy typy zapisów o różnej zawartości informacyjnej dla:

— jednostek organizacyjnych od zakładu do gniazda i linii produkcyjnej
— grup maszyn zamiennych (ogólne normatywy remontowe i eksploatacyjne)
— poszczególnych egzemplarzy maszyn (szczegółowe dane eksploatacyjno-remontowe).

Poza omówioną powyżej zmianą niezbędne jest uzupełnienie zbiorów bazy danych o piąty zbiór, który proponuje się nazywać zgodnie z pełnioną funkcją, kartoteką struktury produkcyjnej. Zbiór ten powinien być zorganizowany



Rys. 2. Baza danych systemu BOMP (zmodyfikowana) — strzałki symbolizują łańcuchowe powiązania adresowe

na tych samych zasadach co kartoteka struktury wyrobów i zawierać np. informacje o tym, jakie:

— gniazda i linie produkcyjne zorganizowano w oddziałach produkcyjnych

— grupy maszyn zainstalowano w gniazdach i liniach

— egzemplarze maszyn tworzą grupę maszyn zamiennych,

a także pozostałe informacje o strukturze produkcyjnej przedsiębiorstwa. Zapisy kartoteki struktury produkcyjnej powiązane będą łańcuchami adresowymi z odpowiednimi zapisami kartoteki jednostek produkcyjnych, która obok swych podstawowych funkcji pełnić będzie rolę kartoteki głównej pozycji.

Jeżeli więc w zapisach egzemplarzy maszyn kartoteki jednostek produkcyjnych zawarte będą szczegółowe dane o terminach wykonania czynności remontowych i ich pracochłonności, to za pomocą programów wykorzystujących powiązania adresowe między tą kartoteką a kartoteką struktury produkcyjnej, będzie możliwe uzyskanie dokładnych danych o dysponowanym czasie pracy w każdej jednostce produkcyjnej, a następnie porównanie ich z danymi o planowanym obciążeniu.

Proponowana metoda wymaga dokonania pewnych zmian w standardowych programach firmy IBM służących do zakładania bazy danych systemu BOMP. Możliwości takie istnieją, gdyż zmiany te zostały dokonane przez zespół Ośrodka Elektronicznego Przetwarzania Danych Fabryki Osprzętu Samochodowego w Łodzi. Działanie podstawowych programów systemu zostało po modyfikacji przetestowane na danych modelowych, a w chwili obecnej trwają prace nad przystosowaniem systemu do zastosowania w praktyce.

WERNER KUNITZ
VEB LEUNA-WERKE WALTER ULBRICHT
Leuna
Niemiecka Republika Demokratyczna

Zautomatyzowany system zarządzania kombinatem LEUNA–WERKE WALTER ULBRICHT

Omówiono stan zastosowań informatyki w kombinacie, dziedziny zastosowań (produkcja, gospodarka materiałowa, zbyt, zatrudnienie, środki trwałe, transport, finanse) oraz dalsze kierunki rozwoju zastosowań EPD.

Rozwój zastosowań elektronicznej techniki obliczeniowej

W kombinacie LEUNA-WERKE WALTER ULBRICHT w latach sześćdziesiątych wykorzystywano do usprawnienia procesów obliczeniowych maszyny licząco-analityczne. Przez wprowadzenie komputera ZRA1 w roku 1961 zgromadzono pewne doświadczenia w zakresie stosowania elektronicznej techniki obliczeniowej, szczególnie w dziedzinie obliczeń naukowo-technicznych.

Zastosowanie sprzętu drugiej generacji dało w połowie lat sześćdziesiątych początek automatyzacji procesów planowania, zarządzania i rozliczeń. W tej fazie rozwijano przede wszystkim projekty zastosowania elektronicznego przetwarzania danych do prac rozliczeniowych.

Przygotowanie zastosowań rozpoczęto od podstawowych obliczeń w dziedzinie produkcji, obrotu materiałowego, środków trwałych i płac, a kontynuowano w dziedzinie rozliczania kosztów i obliczeń finansowych. Drugim punktem węzłowym było planowanie produkcji. Po wdrożeniu szeregu projektów EPD uwidoczniły się istotne efekty usprawniające działalność przedsiębiorstwa.

Zapotrzebowanie na moc obliczeniową stale wzrastało, tak, że po około pięciu latach konieczne stało się rozszerzenie istniejącej mocy obliczeniowej przez zainstalowanie maszyny trzeciej generacji. Nowymi punktami węzłowymi zastosowania EPD stały się:

— wszechstronne wykorzystanie w procesach planistycznych produkcji, zaopatrzenia, zbytu i gospodarki remontowej

— większe wykorzystanie w dziedzinie operatywnego kierowania i zarządzania

— usprawnienie udostępniania informacji naukowo-technicznej.

Stan zastosowań informatyki

Ogólna charakterystyka

Do chwili obecnej procesy planowania, zarządzania i rozliczeń oparto w kombinacie na dziewięćdziesięciu czterech projektach z dziedziny produkcji, obrotu materiałowego, zbytu, transportu, zatrudnienia, środków trwałych i gospodarki finansowej.

Tak szerokie zastosowanie informatyki umożliwiło zmniejszenie stanu zatrudnienia w sferze pracy administracyjnej. Roczne korzyści zastosowania informatyki przekraczają roczne nakłady ponad dwa i pół raza. Korzyści ujawniają się np. we wzroście produkcji, obniżeniu norm zużycia materiałowego i energii, zmniejszeniu zapasu środków obrotowych oraz różnic inwentaryzacyjnych.

Ogólna struktura zapotrzebowania czasu maszyny na realizację zastosowań EPD w dziedzinie planowania i zarządzania przedstawia się w kombinacie następująco:
operatywne zarządzanie i kierowanie — 23%
planowanie i bilansowanie — 21%
rozliczanie — 56%.

Dziedziny zastosowań

Kompleks projektowy „produkcja"

Opracowano i wdrożono projekty EPD dla następujących zagadnień z dziedziny planowania, zarządzania i rozliczeń:
— **roczne planowanie produkcji.** Na podstawie norm zużycia materiałów, możliwości wykorzystania mocy urządzeń produkcyjnych, możliwości surowcowych i możliwości zbytu przeprowadza się obliczenie optymalizacyjne. Optymalne plany produkcyjne ustalone są tak, by uwzględniając założone ograniczenia uzyskać dla kombinatu maksymalny zysk planowy. Określa to jednocześnie profil zdolności produkcyjnych kombinatu. W konsekwencji optymalizacji następuje bilansowanie umożliwiające sporządzenie bilansów materiałowych i energetycznych.
Na zakończenie bilansowania powstaje plan produkcji w układzie wg wyrobów gotowych.
— **długoterminowe planowanie produkcji.** Projekt ten zbudowany jest na tych samych zasadach, jak projekt „Roczne planowanie produkcji", jednak informacje wejściowe są mniej szczegółowe
— **dzienne raporty z wykonania planu produkcyjnego.** Projekt ma na celu usprawnienie informacji przeznaczonej dla kierownictwa kombinatu; zapewnia najkrótszą drogę sprawozdawczości dziennej z uwzględnieniem jej maksymalnej dokładności.

Sprawozdania dotyczą wybranych informacji. Tak więc przykładowo ujmuje się te wyroby gotowe, których wartość stanowi łącznie około 95% wartości produkcji towarowej kombinatu. Dla nich podaje się dzienną wielkość produkcji globalnej i towarowej, a dla produkcji towarowej także narastająco wartość od początku miesiąca i od początku roku.

Oprócz tego podawane jest zużycie i zapasy najważniejszych surowców, półproduktów i wyrobów gotowych.

— **Dekadowe rozliczanie planowanego zysku.** W projekcie tym ujęte są informacje o wykonaniu planu rentowności całego kombinatu i poszczególnych zakładów celem przygotowania operatywnych decyzji kierowniczych. Wykaz wykonania planu rentowności powstaje co dekadę w sposób narastający, zawsze od początku miesiąca.

— **Zestawienie zapasów surowców i wyrobów gotowych.** Projekt ten zapewnia otrzymywanie aktualnego przeglądu zapasów najważniejszych surowców i wyrobów gotowych dla okresów bieżących i ubiegłych. Obok zmniejszenia udziału prac powtarzalnych projekt służy głównie do szybkiego rozpoznawania słabych punktów i rezerw w procesie produkcyjnym, poprawy dyspozycyjności zbytu i ustalenia możliwości zmagazynowania w przypadku nadprodukcji. Podawane są maksymalne zdolności magazynowania, stany alarmowe zapasu, normy zapasów, efektywny zapas dzienny oraz dzienna zdolność magazynowania. Informacje tworzone są dekadowo i udostępnione na żądanie.

— **Miesięczne rozliczanie produkcji chemicznej i energii.** Dla produktów chemicznych opracowywany jest łączny bilans w ujęciu ilościowym i wartościowym. Bilansowanie przeprowadzane jest miesięcznie i narastająco od początku roku.

Półprodukty rozliczane są według kosztów własnych produkcji, natomiast wyroby gotowe według kosztów własnych całkowitych. Z projektu tego wynikają również ilości i wartości dla projektu „Rachunek kosztów".

— **Krótkoterminowe ewidencjonowanie, rozliczanie i bilansowanie produkcji chemicznej i środków energetycznych.** Projekt ten poprzedza projekty: „Dzienny raport o wykonaniu planu produkcji" i „Miesięczne rozliczenie produkcji chemicznej i energii". Jako dane wyjściowe wykorzystane są głównie ręcznie wyznaczane wartości diagramowe. W wyniku obliczeń projektowych ustalane są wielkości przepływu produktów gazowych. Na podstawie tych obliczeń wylicza się i wyprowadza specyficzne dla kombinatu wskaźniki, wartości średnie i zestawienia procentowe. Dane te ujęte są w tabelach i służą określonym obszarom działalności kombinatu jako narzędzia operatywnego kierowania w zakresie koordynacji produkcji. Obliczenia realizowane są codziennie.

- Poszczególne działy produkcji wykorzystują określone projekty EPD jako narzędzia operatywnego kierowania produkcją, np.:
— automatyczne śledzenie „wąskich gardeł". Codziennie ustala i podaje się dla każdego wydziału produkcyjnego faktyczną i maksymalnie możliwą do realizacji produkcję zgodnie z faktycznym przebiegiem procesu wytwórczego poszczególnych produktów.
— sprawozdawczość i analiza wskaźników jakościowych
— szybka informacja ekonomiczna.

Do kontroli i zabezpieczenia wykonania planu określonej dziedziny produkcji wykorzystywane są informacje dekadowe o wykonaniu planu wybranych produktów, o obniżce kosztów materiałowych oraz wykonaniu planu rentowności w każdym z wydziałów produkcyjnych.

Operatywne ustalanie wytycznych. Dla poprawy warunków przygotowania i podejmowania decyzji na stanowiskach dyspozytorskich określonej dziedziny produkcji wylicza się co tydzień lub na żądanie wielkości dla wytycznych produkcji. Wielkości te wykazywane są w poszczególnych „wąskich gardłach" w postaci wytycznych dotyczących rezerw mocy, zużycia surowców, zapasów i wyników produkcyjnych.

Kompleks projektowy „gospodarka materiałowa"

W dziedzinie gospodarki materiałowej wdrożono EPD dla materiałów pomocniczych w zakresie ewidencjonowania i analizy zapasów, rozliczeń, sprawozdawczości i analizy rozchodu materiałów, planowania materiałowego, dysponowania materiałami i realizacji zamówień.
Daje to znaczne korzyści kombinatowi, głównie w wyniku zmniejszenia zapasów ponadnormatywnych, kosztów magazynowania, różnic inwentaryzacyjnych i płaconych odsetek. Dzięki przejęciu przez EPD prac powtarzalnych, jak również wprowadzeniu nowych form organizacyjnych stworzono warunki lepszego wykorzystania personelu.
Dotychczas w dziedzinie gospodarki materiałowej opracowano i wdrożono poniżej podane projekty. Projekty te przewidują cykle przetwarzania dzienne, dekadowe, miesięczne lub na żądanie.
— Tworzenie, zmiany i ujmowanie nowych pozycji w zakresie danych stałych kartoteki materiałowej.
Struktura danych stałych kartoteki materiałowej umożliwia swobodny dostęp do każdego wyrobu. Indeksowo-sekwencyjna metoda zapamiętywania danych stałych i danych o zapasach następuje wg symboli materiałów i kont magazynowych. Projekt ten jest podstawą następujących projektów kompleksu gospodarki materiałowej.
— Przegląd ewidencji materiałowej. Codziennie następuje przegląd stanu zapasów z podaniem odchyleń od norm zapasu dla głównych materiałów wg zasad sprawozdawczości o sytuacjach nietypowych. Przegląd ten sporządzony jest wg różnych szczebli zarządzania
— Analiza zapasów materiałowych. Analizę zapasów sporządza się dla każdego symbolu materiałowego z podaniem stanów na poszczególnych kontach magazynowych wg określonego sposobu wydruku. Zapasy materiałowe wykazuje się tutaj jako informacje dla potrzeb kierownictwa. Ponadto zrealizowano następujące projekty kompleksu gospodarki materiałowej:

- rozliczenia materiałowe
- sprawozdawczość materiałowa
- przygotowanie i udostępnianie danych dla innych kompleksów projektowych
- kontrola i analiza występujących zakłóceń rezerw zapasów
- rozliczanie nieodpłatnych dostaw materiałów
- wykaz przychodów materiałów podstawowych
- rozliczanie innych zapasów magazynowych
- prowadzenie inwentaryzacji ciągłej i wykazywanie różnic
- prognozowanie zapotrzebowania materiałów pomocniczych.

Projekty kompleksu gospodarki materiałowej dostarczają odpowiednio zestawione syntetyczne informacje dla kadry kierowniczej oraz informacje dla operatywnej dyspozycji materiałowej odnoszące się do poszczególnych stanowisk pracy.

Kompleks projektowy — „Zbyt"

W kompleksie — „Zbyt" zrealizowane są następujące projekty:

— dzienne rozliczanie zbytu. W projekcie tym dane źródłowe z dokumentów dostawy są tak opracowywane, aby powstały następujące informacje wyjściowe: rachunki i specyfikacje, dowody bankowe, dzienne dane z realizacji, dzienne należności

— porównanie aktualnego stanu kontraktów
— roczna analiza zbytu wg produktów i odbiorców
— reklama — projekt ten ma zagwarantować ukierunkowaną informację reklamową dla dotychczasowych i potencjalnych odbiorców względnie zainteresowanych w celu minimalizacji wysokości nakładów kosztownych prospektów reklamowych
— rozliczenie świadczeń pomocniczych i dodatkowych.

Kompleks projektowy — „Zatrudnienie"

Zadaniem tego kompleksu jest obok obliczania wynagrodzeń wszystkich pracowników kombinatu, obliczanie funduszu płac, zatrudnienia i funduszu czasu pracy łącznie z przeniesieniem na rachunek kosztów. Powstałe stąd informacje dla potrzeb kierownictwa pozwalają podejmować decyzje oraz dokonywać analiz w zakresie problemów ekonomiki pracy i polityki kadrowej.

Dotychczasowe zastosowanie EPD w tej dziedzinie można ująć w dwóch zespołach problemów:

— obliczanie zatrudnienia i wynagrodzeń. Na podstawie kartoteki danych stałych oraz ewidencji czasu pracy i przestojów oraz danych wynikowych innych dziedzin, następuje szczegółowe obliczanie płacy na każdego zatrudnionego. Obliczanie to przeprowadza się poprzez rachunek brutto i netto aż do sporządzenia samej listy płac w podziale według jednostek organizacyjnych kombinatu z uwzględnieniem obliczania wszystkich potrąceń, takich jak czynsz, ubezpieczenie i przelewy bankowe. W wyniku obliczeń indywidualnych płac sporządza się wykaz zbiorczy wszystkich składników płac według klasyfikacji rachunkowości. Informacje są zestawiane tak, aby umożliwiały wykorzystanie do analiz w zakresie ekonomiki pracy oraz do sprawozdawczości wewnątrzzakładowej i zewnętrznej. Zawierają one najważniejsze wskaźniki dotyczące zatrudnienia, funduszu płac i wykorzystania czasu pracy, podzielone według kategorii i grup pracowników. Równocześnie rejestrowany jest ruch płac w kombinacie i udostępniony dla potrzeb rachunku kosztów w podziale wg rodzaju kosztów

— ilościowe i jakościowe zestawienie stanu zatrudnienia. W ramach coraz lepszego wykorzystania społecznych zasobów pracy przygotowano w tym projekcie informacje dla kierownictwa, umożliwiające dokładne i ciągłe planowanie rozwoju zatrudnienia. W konsekwencji informacje zawarte w projekcie umożliwią centralną kontrolę i analizę ogólnego rozwoju zatrudnienia oraz skuteczne przygotowanie decyzji ekonomicznych dla bardziej efektywnego wykorzystania pracowników kombinatu.

Projekt składa się z następujących części:

- obliczanie i kontrola stanu zatrudnienia z punktu widzenia polityki kadrowej. Tutaj dokonuje się m.in. analizy ruchu pracowników według przyczyn fluktuacji, wykaz zatrudnienia według rodzaju pracy, wieku, zawodów i poziomu kwalifikacji.
- zestawienie rozwoju zatrudnienia oraz selekcji kadr.

Kompleks projektowy — „Środki trwałe"

W dziedzinie gospodarki środkami trwałymi opracowano i wdrożono następujące projekty EPD:

— rozliczanie środków trwałych. Projekt obejmuje miesięczne rozliczanie oraz sporządzanie zestawień statystycznych dotyczących nabycia i skreślenia środków trwałych oraz ich przemieszczania wewnątrz kombinatu, a także wyliczenie odpisów amortyzacyjnych i aktualnej wartości księgowej tych środków

— planowanie środków trwałych. Projekt obejmuje planowanie przychodów środków w wyniku inwestycji, planowanie spodziewanej amortyzacji, planowanie przewidywanych odpisów amortyzacyjnych. Dalej dokonuje się porównania stanu istniejącego z planowanym w zakresie przychodu i rozchodu środków oraz planowanej amortyzacji

— planowanie inwestycji. Projekt obejmuje planowanie inwestycji według zleceniobiorców, pozycji bilansowych i struktury urządzeń oraz porównywanie planu ze stanem istniejącym w celu kontroli realizacji inwestycji

— planowe zabezpieczanie eksploatacji środków trwałych. Projekt obejmuje rejestrowanie planowo przeprowadzanych zabiegów zabezpieczających, niezbędne bilansowanie dysponowanych mocy przerobowych na remonty i konserwacje, planowanie materiałów potrzebnych dla zabezpieczenia eksploatacji środków trwałych oraz wystawianie dokumentacji zleceń (dowody płacowe, dowody pobrania materiałów)

— rozliczanie zleceń czynności warsztatowych i pozostałych czynności pomocniczych. Projekt obejmuje rejestrację usług własnych warsztatów (moce dla zabezpieczenia eksploatacji, usługi dla obcych instytucji oraz dla inwestycji), własnych mocy transportowych, przerobu własnych biur projektowych i konstrukcyjnych, ujętych umowami usług obcych i zużycia materiałów pomocniczych łącznie z wywartościowaniem tych usług oraz obciążaniem kontami zleceń kont zleceniodawców.

Kompleks projektowy — „Transport"

W kompleksie tym zrealizowano następujące projekty:

— ekspedycja towarów. Ujmuje codziennie przesyłki towarowe, wychodzące. Uwzględnia dzienne uzgodnienie kosztów przewozowych z koleją, dzienne przygotowanie dowodów i statystyk wysyłkowych, obliczanie i sporządzanie zestawień należności i frachtów oraz ich przenoszenie na rachunek kosztów, a także rozliczenia finansowe.

— rozliczanie transportu wewnątrzzakładowego. Obejmuje ewidencjonowanie kosztów i wydajności różnych środków transportu kombinatu, wyliczenie wydajności na jeden pojazd oraz przekazywanie tych danych na rachunek kosztów i zleceń.

Kompleks projektowy — „Finanse"

W kompleksie tym opracowano i wdrożono następujące projekty:

— Rachunek kosztów wg rodzajów i miejsc powstawania. Wykonuje się:
- zintegrowane przetwarzanie danych, które w opisanych projektach obliczeń podstawowych i szczegółowych zostały zarejestrowane oraz przystosowane do wykonania rozliczeń
- zarejestrowanie i opracowanie danych zgodnie z ramowym planem kont
- kombinowany rachunek kosztów, nakładów i wyników według miejsc powstawania kosztów i dla określonych zakresów działalności, w zakładzie, względnie w kombinacie, dla okresów miesięcznych i narastająco
- porównanie planu i wykonania dla zaangażowanych oraz wszystkich pozostałych miejsc powstawania kosztów
- wykaz wyników księgowych według ustalonych wskaźników dla użytku zespołów wykonawczych

— Rozliczanie nośników kosztów obejmujące:
- opracowanie danych odpowiadające istniejącym potrzebom kalkulacyjnym
- wykaz faktycznego kształtowania się kosztów własnych według nośników oraz ich porównanie z planem według określonych składników kosztów
- uzyskanie podstaw rozrachunkowych dla planowania i rozliczania podstawowych nakładów dla każdego następnego okresu roboczego
- uzyskanie wielostopniowego rachunku kosztów i kalkulacji do kontroli kalkulacji cen, obliczenia skutków zmian cen i otrzymywania danych dla potrzeb przepływów międzygałęziowych gospodarki narodowej

— Planowanie kosztów i wyników. Projekt ten pozwolił osiągnąć:
- zintegrowane przetwarzanie danych planistycznych innych kompleksów projektowych
- planowanie w oparciu o normatywy tych kosztów i nakładów, które nie są przedmiotem planowania materiałowego
- kombinowany plan kosztów, nakładów i wyników według miejsc powstawania kosztów dla wyższych szczebli zarządzania aż do poziomu zakładu lub kombinatu

— wykaz zaplanowanych przyrostów zysku według miejsc powstawania kosztów dla wyższych szczebli zarządzania w porównaniu z planem i(lub) wykonaniem poprzedniego roku, według elementów decydujących

— ewidencja rachunków bieżących. Projekt umożliwia:
- maszynową rejestrację należności według terminów płatności i przygotowanie danych dla podejmowania operatywnych dyspozycji finansowych
- maszynowe wyrównywanie należności przez napływające wpłaty i uznania wg numerów rachunków
- maszynowe sporządzanie monitów z wyliczaniem odsetek za zwłokę
- maszynowe zestawianie należności do sporządzania potwierdzeń sald odbiorców na określony dzień
- przekazywanie sum dziennikowych do księgowości finansowej dla prowadzenia kont syntetycznych
- maszynową rejestrację zobowiązań według terminów płatności dla operatywnych dyspozycji finansowych
- maszynowe wyrównywanie zobowiązań przez dokonywane zapłaty i obciążenia według numerów rachunków
- przekazywanie sum dziennikowych, szczególnie kont

wpływu rachunków dla potrzeb prowadzenia kont syntetycznych
- przekazanie do księgowości finansowej pojedynczych pozycji ujętych na kontach wpływu rachunków wg numerów rachunków

— Księgowość finansowa
Obejmuje:
- przygotowanie danych do analitycznego wykazu kont finansowych
- listę sald dotyczących miesięcznych stanów początkowych, obrotów i stanów końcowych kont finansowych
- kumulowanie obrotów kont finansowych do maszynowego prowadzenia księgi głównej
- uzyskanie danych do bilansu i rachunku wyników oraz sporządzenia sprawozdawczości finansowej.

Dalszy rozwój zastosowań EPD

Dalszy rozwój zastosowań EPD w kombinacie będzie ukierunkowany następująco:
— włączanie nowych dziedzin zastosowań. Istnieje zadanie rozszerzenia EPD dla potrzeb intensyfikacji produkcji, głównie w dziedzinie operatywnego kierowania procesami produkcji, zabezpieczenia mocy przerobowych i zaopatrzenia. Z drugiej strony należy w szerokim zakresie wykorzystywać EPD do planowania rocznego i długoterminowego
— usprawnienie udostępniania informacji dla kierownictwa i pracowników.

Usprawnienie udostępniania informacji winno być osiągane przez rozwijanie informacji kompleksowych z tworzonych informacji jednostkowych różnych projektów.

Informacje takie będą celowo gromadzone i ukierunkowywane stosownie do potrzeb różnych szczebli zarządzania. Obejmuje to również analizę określonych kierunków rozwoju w oparciu o analizy odpowiednich czynników tego rozwoju.

Udostępnianie informacji należy przy tym tak zorganizować, aby oddać je dyspozycji kierownictwa i pracowników w możliwie krótkim okresie czasu.

Osiągnąć to można z jednej strony przez skrócenie czasu na przygotowanie danych źródłowych, a z drugiej strony umożliwienie poprzez dialogi z maszyną bezpośredniego dostępu do informacji.

Ważne zadanie usprawnienia dostępu do informacji polega również na tym, że maksymalnie ograniczy się obowiązkowe wyprowadzanie informacji według rozwiązania projektowego, które często nie wynika z faktycznego zapotrzebowania na te informacje.

Chodzi tutaj o bardziej celowe ukierunkowanie przygotowania informacji w projektach dla poszczególnych zadań kierowania, planowania i rozliczania, aby nie wynikały one tylko z rozwiązania projektowego, lecz aby były udostępniane zainteresowanym tylko w zależności od konkretnego zapotrzebowania.

Omówione wyżej punkty ciężkości problemu wymagają szczególnie głębokiej integracji zarówno kompleksów projektowych, jak i poszczególnych projektów.

Niektóre zadania można będzie zrealizować jednak tylko wtedy, gdy koncepcja większości projektów zostanie przemyślana od nowa. Z tego powodu zaprezentowane kompleksy projektowe zostaną na nowo opracowane wspólnie z innymi kombinatami, celem stworzenia podstawy dla wzrostu efektywności zautomatyzowanego systemu zarządzania.

ANDRZEJ SOKOŁOWSKI
Zakład Informatyki
Wojskowej Akademii Politycznej
Warszawa

Skuteczność metod ochrony zbiorów informacji w systemach informatycznych

Omówiono skuteczność ochrony zbiorów na przykładzie kilku metod ochrony z grupy metod programowych i technicznych i podano czynniki warunkujące skuteczność ochrony informacji.

Przez skuteczność metody ochrony rozumie się stopień, w jakim dana metoda zabezpiecza zbiory informacji przed infiltracją lub przypadkowym zniszczeniem. Wynika z tego, że skuteczność jest podstawowym czynnikiem decydującym o ocenie walorów danej metody ochrony, a tym samym służy za zasadnicze kryterium wyboru metody.

Określenie skuteczności metody w przypadku ochrony zbiorów informacji przed zniszczeniem przez żywioł lub przypadkowym zniszczeniem podczas niewłaściwej pracy personelu, sprowadza się w zasadzie do stwierdzenia czy ochroni ona zbiory przed zniszczeniem, czy też nie. Wszelkiego rodzaju stany pośrednie, które nie eliminują możliwości częściowego zniszczenia zbioru informacji, z punktu widzenia użyteczności takiego zbioru w dalszym procesie przetwarzania — wskazują na małą przydatność takiej metody. Należy przy tym dodać, że określenie skuteczności ochrony metod organizacyjnych, w których decydujące znaczenie ma czynnik ludzki, jest o wiele trudniejsze od pozostałych.

Metody ochrony programowe i techniczne (jak też zastosowanie szyfrów) oceniamy pod kątem możliwości zaistnienia infiltracji. Takie postępowanie jest konsekwencją tego, że skuteczność ochrony jest tym większa, im mniejsza jest możliwość infiltracji lub zniszczenia zbioru informacji.

Skuteczność ochrony zbiorów informacji

Miarą skuteczności jest prawdopodobieństwo ochrony zbioru informacji przy użytej metodzie ochrony. Przy czym prawdopodobieństwo to określa się pośrednio poprzez wielkość prawdopodobieństwa infiltracji lub zniszczenia zbioru informacji, jakie może zaistnieć przy zastosowaniu metody ochrony.

Prawdopodobieństwo ochrony zbioru oznaczymy przez $P(S)$, natomiast możliwość obejścia metody, a tym samym prawdopodobieństwo infiltracji, przez $P(R)$. Zakładamy więc, że

w przypadku braku skutecznej ochrony, możliwość infiltracji jest równoznaczna z jej realizacją. Wobec tego zdarzenia skutecznej ochrony S i infiltracji R są zdarzeniami przeciwnymi. Jest to słuszne, ponieważ brak skutecznej ochrony jest — przy tym założeniu — równoznaczny z infiltracją zbioru, co możemy zapisać następująco:

$$P(S) + P(R) = 1,$$

a stąd prawdopodobieństwo skutecznej ochrony zbioru wynosi:

$$P(S) = 1 - P(R) \tag{1}$$

Z powyższego wynika, że dla określenia prawdopodobieństwa ochrony zbioru $P(S)$, należy określić prawdopodobieństwo infiltracji $P(R)$. Ustalenie tego prawdopodobieństwa wymaga dokonania wielu doświadczeń (na ogół poza przypadkami tych metod, przy których skuteczność można określić wzorami przed ich zastosowaniem), które polegają na stwarzaniu sytuacji zagrożenia zbioru informacji z jednoczesnym badaniem zachowania się metody ochrony. Materiał z takich doświadczeń, może służyć do określenia prawdopodobieństwa ochrony zbioru przy zastosowaniu danej metody, jak też innych jej właściwości.

Prowadząc obserwację n wyłączających się doświadczeń, gdzie n to liczba przypadków skutecznej ochrony i infiltracji, podczas zagrożenia zbioru informacji, stwierdzamy, że zaistniało m przypadków realizujących zdarzenie infiltracji R. Wobec tego prawdopodobieństwo infiltracji $P(R)$ wynosi:

$$P(R) = \frac{m}{n} \tag{2}$$

Wynika stąd, że prawdopodobieństwo infiltracji $P(R)$ równe jest ilości przypadków sprzyjających zdarzeniu rozszyfrowania R do ilości wszystkich przypadków zaistniałych podczas badania [1]. Korzystając zaś ze wzorów (1) i (2), można określić prawdopodobieństwo ochrony zbioru $P(S)$.

Skuteczność grup metod ochrony

Skuteczność ochrony zbiorów informacji zostanie omówiona na przykładzie kilku metod ochrony z dwóch grup (metody programowe i techniczne). Należy jednak zastrzec, że przy rozpatrywaniu grup metod ochrony, możemy mówić jedynie o tym, że dana grupa na ogół jest bardziej lub mniej skuteczna od innej grup metod. Wynika to z faktu, że w zasadzie, każda grupa metod chroni informacje na innym etapie przetwarzania. Ponadto przeważnie każda metoda ochrony w danej grupie różni się od innej nie tylko zasięgiem ochrony, ale też poziomem jej skuteczności. Stąd wniosek, że **skuteczność metody ochrony należy określać dla każdej metody oddzielnie.**

Skuteczność programowych metod ochrony

Skuteczność omówionych programowych metod ochrony (por. [2]) można określić na podstawie obliczenia prawdopodobieństwa ochrony zbiorów informacji. Wcześniej wspomniano, że w celu obliczenia prawdopodobieństwa ochrony zbioru informacji, należy przeprowadzić doświadczenie, np. na zbiorze modelowym, polegające na badaniu sytuacji zagrożenia tego zbioru. Przebieg takiego doświadczenia możemy nazwać procesem zagrożenia zbioru.

Przedstawiony poniżej model takiego procesu, który symuluje zagrożenie zbioru, wymaga opisu [3]. Załóżmy więc, że algorytm $M = (Z; F)$ reprezentuje strukturę relacyjną dla Z (zbiór stanów — wyników doświadczenia) i operacji jednoargumentowej F w Z (funkcja przejścia). Funkcję przejścia (sprowadzenie tożsamości) określimy następująco:

$$F(k, d_n) = \begin{cases} k, \text{ gdy } k = d_k, \\ d_n, \text{ gdy } k \neq d_n \quad \text{(przejście do ponownej operacji badania)}, \end{cases}$$

gdzie:

k — kod ochrony,
d — liczba (słowo) spełniające rolę kodu w procesie symulacji.

Można więc stwierdzić, że algorytm M jest przepisem przekształcenia określonych danych (stany) według stałej reguły (funkcji przejścia) i za pomocą określonych środków. A zatem procesem algorytmu $M = (Z; F)$ nazywamy ciąg $d_0, d_1, d_2, \ldots$, stanów (wyników doświadczenia) ze zbioru Z.

Obliczeniem algorytmu $M = (Z; F)$ nazywamy **proces tego algorytmu** postaci d_0, $d_1 = F(d_0)$, $d_2 = F(d_1)$, ..., który jeśli jest skończony, kończy się stanem d_k (końcowym), nie należącym do dziedziny operacji F. Obliczenie d_0, $d_1 = F(d_0)$, $d_2 = F(d_1)$, ..., jest wyznaczone przez algorytm M i przez stan d_0 (początkowy).

Stan początkowy d_0 stanowi dane początkowe, stany d_0, $d_1 = F(d_0)$, $d_2 = F(d_1)$, ..., stanowią wyniki pośrednie, a stan końcowy d_k (o ile istnieje) — wynik końcowy procesu symulacji zagrożenia zbioru.

W powyższym opisie nie uwzględniono parametru czasu t, ponieważ czas procesu symulacji jest zdeterminowany bądź przez osiągnięcie stanu końcowego d_k, bądź też przez prowadzącego obserwację stanów zagrożenia.

Czas t określony jest przedziałem $\langle t_0, t_n \rangle$, przy czym chwila początkowa t_0 reprezentuje rozpoczęcie procesu, natomiast t_n może być wyznaczona przez:

$$t_n = \begin{cases} t_k \text{ — osiągnięcie stanu } d_k, \\ t_n \text{ — określony przez prowadzącego obserwację.} \end{cases}$$

W oparciu o powyższe możemy przykładowo obliczyć prawdopodobieństwo skuteczności ochrony metody programowej, opartej na zapisie kodu dostępu w etykiecie zbioru informacji znajdującego się na taśmie magnetycznej. Załóżmy, że kod dostępu jest liczbą czterocyfrową, a metoda ochrony polega na badaniu prawa dostępu użytkownika do określonego zbioru. Zezwolenie na korzystanie ze zbioru odbywa się na podstawie porównania kodu zapisanego w etykiecie zbioru z kodem podanym przez użytkownika na karcie dziurkowanej, odczytanym przez czytnik kart. W przypadku zgodności tych kodów, program użytkownika zostaje dopuszczony do korzystania ze zbioru. W przeciwnym przypadku — program ochrony podaje na pulpicie operatora komunikat stwierdzający, że użytkownik nie posiada upoważnienia do korzystania z danego zbioru.

W omawianym przykładzie przyjęto 250 kart maszynowych z wydziurkowanymi liczbami czterocyfrowymi, wybranymi w sposób losowy. Badanie skuteczności metody przeprowadza się na zastrzeżonym zbiorze modelowym z zapisanym kodem w etykiecie tego zbioru. Do pamięci operacyjnej wczytuje się program modelowy, którego zadaniem jest odczytanie zbioru zastrzeżonego. Następnie do czytnika kart wkłada się zbiór kart z wydziurkowanymi liczbami, mającymi spełnić rolę kodów dostępu. Po rozpoczęciu pracy przez program, obserwuje się pulpit operatora, licząc ilość komunikatów określających brak dostępu.

Program modelowy jest tak zbudowany, że po każdorazowym otrzymaniu dostępu do zbioru, wypisuje na drukarce jeden rekord informacji, zamyka zbiór i wykonuje skok do początku programu. Następnie praca programu rozpoczyna się ponownie, a więc badany jest następny klucz dostępu. Wyczerpanie się kart z kodami dostępu sygnalizowane jest odpowiednim komunikatem (ze względu na małą ilość słów, jaką można zapisać w etykiecie zbioru, komunikat ma postać skrótu literowego lub umowną liczbę) na pulpicie operatora. W tym momencie następuje zakończenie badania. Można więc przystąpić do zliczenia na wydruku z pulpitu operatora ilości komunikatów stwierdzających brak dostępu programu modelowego do zastrzeżonego zbioru. Załóżmy, że otrzymaliśmy 246 odpowiedzi negatywnych. Wobec tego ilość zdarzeń sprzyjających infiltracji wynosi $250 - 246 = 4$, co sprawdzamy z ilością wydrukowanych rekordów na drukarce wierszowej. Na podstawie wzoru (2) otrzymujemy:

$$P(R) = \frac{4}{250} = 0{,}016.$$

Stąd prawdopodobieństwo ochrony zbioru programowej metody ochrony z kodem zapisanym w etykiecie zbioru, obliczone na podstawie wzoru (1), wynosi:

$$P(S) = 1 - 0{,}016 = 0{,}984.$$

Takie prawdopodobieństwo skuteczności ochrony jest bardzo wysokie. Natomiast teoretyczne prawdopodobieństwo

skuteczności metody, obliczone na podstawie ilości możliwych kluczy czterocyfrowych, jest jeszcze wyższe i wynosi 0,9999 [1]).

Biorąc pod uwagę możliwość zaistnienia różnych czynników obniżających skuteczność ochrony, które trudno przewidzieć, wskazane jest przy ocenie metody rozważenie prawdopodobieństwa skuteczności określonego w oparciu o przeprowadzone doświadczenia.

Skuteczność metod technicznych

W porównaniu do metod programowych określenie skuteczności metod technicznych jest nieco trudniejsze. W grupie tych metod (por. [14]) spotykamy się z takimi urządzeniami, jak układy identyfikacji użytkownika, układy zabezpieczające i układy alarmujące, których skuteczność jest różnie określana. Dla każdego z nich skuteczność ochrony można określić następująco:

- układów identyfikacji — na podstawie prawdopodobieństwa możliwości pomyłki układu
- układów zabezpieczających — bądź na podstawie prawdopodobieństwa jego rozszyfrowania (zamki kodowe), bądź przez ocenę jednego z dwóch stanów, tzn. czy układ w danych warunkach zabezpiecza zbiory informacji przed zniszczeniem, czy też nie
- układów alarmujących — na podstawie ich sprawności eksploatacyjnej, przy założeniu, że układy alarmujące są zainstalowane w ośrodku obliczeniowym we właściwych miejscach.

Układy identyfikacji charakteryzują się różną skutecznością ochrony informacji. Najwyższe prawdopodobieństwo skuteczności ochrony informacji posiada układ rozpoznający linie papilarne odcisków palców, wynoszące 0,9995. W układach rozpoznających głos [2]) lub twarz, prawdopodobieństwo skuteczności jest niższe, przy czym nie zależy ono tyle od samej zasady identyfikacji, co od niedoskonałości konwersji głosu lub obrazu na zapis bierny. Natomiast układ identyfikacji użytkownika wykorzystujący kartę z odcinkiem TM, posiada wysokie prawdopodobieństwo ochrony zbiorów. Dla wykonanych obliczeń szacowane było prawdopodobieństwo czynników składowych sprzyjających infiltracji. I na tej podstawie prawdopodobieństwo ochrony zbioru, przy zastosowaniu tego układu, wynosi 0,9941 [3]). Dokładniej prawdopodobieństwo ochrony można ustalić na podstawie obserwacji metody w odpowiednio długim czasie, ponieważ zależy ono przede wszystkim od faktu, czy użytkownik zgubi kartę z odcinkiem TM, kto ją znajdzie (osoba zainteresowana informacjami w systemie lub niezainteresowana, osoba mogąca uzyskać dostęp do urządzenia końcowego) oraz od czasu, po jakim użytkownik zgłosi do ośrodka obliczeniowego fakt zagubienia karty.

Skuteczność układów zabezpieczających można określić na podstawie prawdopodobieństwa skuteczności ochrony informacji obliczonego teoretycznie (np. w przypadku znanej ilości kombinacji kodów) i oszacowanie w oparciu o wyniki obserwacji.

Na przykład prawdopodobieństwo trafienia przez osobę niepowołaną na jeden ustalony kod w zamku kodowo-magnetycznym możemy obliczyć w sposób następujący:

załóżmy, że zamek posiada 9 przycisków ($n = 1, 2, ..., 9$, przycisk 0 służy do kasowania pomyłek), a używany w ośrodku obliczeniowym kod do otwierania zamka jest 4-cyfrowy ($k = 4$). Wobec tego ilość wariacji bez powtórzeń [4]) dla takiego zamka wyniesie:

$$V_n^k = \frac{n!}{(n-k)!} = \frac{9!}{(9-4)!} = 3024.$$

Ponieważ z tej ilości możliwych kodów, tylko jeden jest prawdziwy, (sprzyjający zdarzeniu otwarcia zamka, czyli $m = 1$), to prawdopodobieństwo rozszyfrowania kodu przez osobę niepowołaną, określone na podstawie wzoru (2), wyniesie:

$$P(R) = \frac{1}{3024} = 0{,}0003.$$

Stąd prawdopodobieństwo skuteczności ochrony takiego układu ma wartość:

$$P(S) = 1 - P(R) = 1 - 0{,}0003 = 0{,}9997.$$

Przy tak dużej liczbie różnych kodów, osoba która by chciała otworzyć zamek za pomocą przycisków (zakładając, że na wciśnięcie jednej kombinacji kodu potrzeba 5 s, w tym 4 s na wyciśnięcie samego kodu, a 1 s na wyciśnięcie przycisku 0), w najbardziej niekorzystnym przypadku, potrzebuje 4,2 godz. czasu.

Skuteczność układów alarmujących określić można na podstawie ich sprawności eksploatacyjnej. Innymi słowy można ustalić czy układ jest sprawny i może w określonych warunkach skutecznie funkcjonować, czyli realizować przewidziane funkcje, czy też nie. W jednym z ośrodków miał na przykład miejsce przypadek, że same czujniki przeciwpożarowe były zainstalowane poprawnie i były sprawne lecz zapomniano o przyłączeniu kabli do centralki przeciwpożarowej. Zrozumiałe jest, że w takim przypadku czujniki nie spełniały żadnej roli, a ich skuteczność równała się zeru.

Skuteczność ochrony innych układów możemy ocenić przez określenie jednego z dwóch stanów zachowania się tych układów. W takim przypadku oceniamy czy układ w danych warunkach spełnia wymogi zabezpieczenia zbiorów informacji, czy też nie. Stanów pośrednich układów w zasadzie nie bierzemy pod uwagę. Jeśli np. określamy, że dana szafa metalowa do przechowywania magnetycznych nośników informacji (MNI), spełnia wymogi ochrony przeciwpożarowej i chroni przed działaniem pola magnetycznego (jest ekranowana), to można uznać, że nośniki te są chronione skutecznie podczas magazynowania.

Odnośnie do wszystkich technicznych metod ochrony można postulować, aby producent określał skuteczność ochrony urządzeń wchodzących w ich skład. Natomiast kontrola skuteczności ochrony metod technicznych powinna być przeprowadzana w ośrodku obliczeniowym.

Wydaje się konieczne przestrzeganie warunku, że każda metoda techniczna, jak również pozostałe metody, powinny mieć ustaloną skuteczność przed ich wprowadzeniem do eksploatacji, ponieważ bez znajomości skuteczności ochrony danej metody, nie możemy nic powiedzieć o stopniu ochrony zbiorów informacji.

Czynniki warunkujące skuteczność ochrony informacji

W czasie eksploatacji systemu API, skuteczność ochrony informacji nie jest zawsze stała. Skuteczność ta może ulegać zmniejszaniu lub, podczas pewnej ilości cykli przetwarzania informacji, być utrzymywana na tym samym poziomie.

Do czynników wpływających na działanie metody ochrony, a zmniejszających skuteczność ochrony należą:

— zła organizacja pracy w ośrodku obliczeniowym
— nieprzestrzeganie warunków eksploatacyjnych metody ochrony
— wady techniczne systemu cyfrowego
— wady systemu informatycznego
— nieodpowiednie warunki transportu maszynowych nośników informacji i zestawień wynikowych.

Natomiast do czynności wykonywanych przez personel i użytkowników a wpływających również na zmniejszenie skuteczności ochrony, można zaliczyć:

— niesumienność personelu
— niewykonywanie określonych czynności wymaganych przy stosowaniu metod ochrony, po awariach lub innych przerwach w przetwarzanym systemie
— nieprzestrzeganie wymogów ochrony samej metody, jak zgubienie karty identyfikacyjnej, dopuszczenie do wykonania kopii żetonów, kluczy, **gadulstwo** (ujawnienie struktury metody ochrony lub jej elementu)
— niedocenianie znaczenia ochrony informacji.

---

[1]) Liczba k — wyrazowych wariancji z powtórzeniami ze zbioru o n elementach równa jest n^k, więc dla $n = 0, 1, 2, ..., 9$ i $k = 4$ otrzymujemy ilość kluczy równą 10 000. Stąd teoretyczne prawdopodobieństwo omawianej metody, gdy $P(R) = 0{,}0001$, wynosi $P(S) = 1 - 0{,}0001 = 0{,}9999$.

[2]) Np. istnieje obecnie system pozwalający na sprawdzanie podstawowych cech głosu człowieka za pomocą komputera. Prawdopodobieństwo identyfikacji głosu wynosi 0,988, przy czym czas oczekiwania na wynik identyfikacji wynosi 12 s. [5].

[3]) Prawdopodobieństwo ochrony zbioru przy zastosowaniu układu identyfikacji wykorzystującego kartę z odcinkiem TM, obliczono w podobny sposób, jak dla przykładu metody programowej.

[4]) Ze względu na rozwiązanie techniczne zamka przyjęto ilość wariancji bez powtórzeń, czyli wykluczono takie postacie kodu, w których dana cyfra może się powtórzyć, np. 11*nn*, 222*n*, 3*nn*3.

Do czynników wpływających na utrzymanie poziomu skuteczności ochrony zaliczamy:
— zabezpieczenie samych metod ochrony
— sumienne wykonywanie wszystkich czynności wymaganych przy stosowaniu metod ochrony
— odpowiednio częste zmienianie haseł (kluczy) dostępu w metodach ochrony
— kontrole okresowe i wyrywkowe pracy samych metod ochrony i przestrzegania wszystkich wymogów
— wyznaczenie w ośrodku obliczeniowym osoby, która mając zmniejszone obowiązki służbowe, byłaby odpowiedzialna za całokształt ochrony informacji
— analizowanie wykrytych przypadków infiltracji lub zniszczenia (uszkodzenia) zbiorów informacji (badanie ankietowe wykazuje, że w około 80% ośrodków obliczeniowych w Polsce zaistniały przypadki zniszczeń zbiorów informacji) i zapobieganie im w przyszłości
— zachowanie możliwej anonimowości pracy niektórych ośrodków i przetwarzanych systemów, a szczególnie struktury organizacyjnej tych systemów.

Ostatecznie można stwierdzić, że stabilizacja skuteczności ochrony informacji wymaga w pierwszej kolejności ochrony samej metody, częstej kontroli jej działania i właściwego postępowania personelu. Żadna z możliwych do zastosowania metod nie chroni całkowicie zbiorów przed infiltracją lub zniszczeniem i prawdopodobnie w przyszłości nie powstanie taka metoda. Dlatego, obok istnienia w systemie metod ochrony, potrzebne jest stałe skoncentrowanie uwagi personelu bezpośrednio związanego z przetwarzaniem informacji na ich bezpieczeństwie. Bowiem wzajemne uzupełnianie się tych dwóch elementów, gwarantuje dopiero wysoką skuteczność ochrony zbiorów informacji.

Istotnym zagadnieniem, nie tylko dla zachowania wysokiej skuteczności metod ochrony, ale nawet ich przydatności, jest fakt, że proces ochrony zbiorów informacji nie jest statyczny. Wymaga on, obok śledzenia rozwoju sposobów infiltracji, ciągłego doskonalenia istniejących metod ochrony oraz szukania nowych rozwiązań.

LITERATURA

[1] Czechowski T.: Elementarny wykład rachunku prawdopodobieństwa. Wyd. 3, PWN, Warszawa 1970.
[2] Sokołowski A.: Programowe metody ochrony zbiorów informacji w systemach informatycznych. INFORMATYKA nr 10, 1975.
[3] Winkowski J.: Symulacja algorytmiczna. Centrum Obliczeniowe PAN, Warszawa 1972.
[4] Sokołowski A.: Techniczne metody ochrony zbiorów informacji w systemach informatycznych. INFORMATYKA nr 12, 1975.
[5] Komputerowy system do weryfikacji głosów. ORGANIZACJA I KIEROWANIE. Przegląd Czasopism Zagranicznych, nr 2, 1974.

Jakie będą komputery lat osiemdziesiątych?*) Prognozy rozwoju sprzętu

Ostatnie dwadzieścia lat w historii rozwoju techniki komputerowej charakteryzuje się wzrostem o rzędy wielkości szybkości komputerów, pojemności pamięci i niezawodności. W tym samym czasie równie poważnie zmalały wymiary fizyczne, pobór mocy i koszt.

Wzrost szybkości obliczeń osiągnięto przez zastosowanie nowych elementów i sposobów produkcji oraz dzięki nowym rozwiązaniom architektonicznym systemów komputerowych, takim jak przetwarzanie o organizacji liniowej *(pipeline)*, procesory o organizacji siatkowej, procesory asocjacyjne i systemy pamięci z przeplataniu. Równocześnie nowe koncepcje oprogramowania, takie jak wieloprogramowość i podział czasu *(time-sharing)* spowodowały bardziej efektywne wykorzystanie systemów komputerowych.

Przy potraktowaniu dotychczasowego postępu jako precedensu może wydawać się, że przewidywanie przyszłych udoskonaleń działania będzie wymagało jedynie ekstrapolowania historycznych trendów.

Jednakże każda nowa generacja elementów i nowa technika wytwarzania wykorzystuje charakterystyki elementów bliższe ograniczeń wynikających z praw fizyki, które wyznaczają ich wymiary fizyczne, szybkość przełączania, rozpraszanie mocy i niezawodność. Innowacje architektoniczne zwiększające złożoność prowadzą również do zmniejszającej się wydajności działania. Ze względu na te ograniczenia proste ekstrapolacja na lata 1980 obecnych szybkości wzrostu powoduje ryzyko przewidywania rzeczy fizycznie nierealizowalnych.

Aby uniknąć tych pułapek i jednocześnie uwzględnić stosunkowo dużą rozpiętość okresu, dla którego opracowano prognozy osiągnięć — prawie tak dużą, jak cały poprzedni okres historii techniki komputerowej — przewidywania oparto na założeniu, że obecna technologia ciała stałego półprzewodnikowych układów logicznych i technika wytwarzania układów o dużym stopniu scalenia będą w dalszym ciągu wykorzystywane w latach siedemdziesiątych, a ich fizyczne ograniczenia będą osiągnięte na początku lat dziewięćdziesiątych. Wynikające stąd przewidywania mogą być traktowane jako „wolne od niespodzianek" w tym sensie, że nie zależą od przewrotów technicznych — powstania nowego, obecnie nieznanego elementu lub nowej technologii produkcji.

Z drugiej strony badacze zajmujący się prognozami technicznymi wskazują, że zasada ciągłości rozwoju odnosi się do postępu technicznego — przewroty techniczne zdarzały się w przeszłości i mogą być oczekiwane w przyszłości. Aby to odrzucić, trzeba by przyjąć tezę, która nie sprawdziła się wiele razy w przeszłości. Wynika z tego oczywiście, że zapotrzebowanie na technikę będzie istniało nadal. Jeśli tak jest, to przedstawione przewidywania będą zbyt zachowawcze i mogą ukazywać tylko dolną granicę pełnych możliwości systemów komputerowych w latach osiemdziesiątych.

Tendencje rozwoju elementów układów logicznych

Technika komputerowa wkracza obecnie w czwartą generację elementów służących do ich budowy — układy logiczne ciała stałego o średnim i dużym stopniu scalenia (odpowiednio: MSI — *medium-scale integration* i LSI — *large-scale integration*). Poprzednie generacje elementów to: przekaźniki, lampy elektronowe, dyskretne tranzystory i scalone układy tranzystorowe.

*) Opracowano na podstawie artykułu R. Turn: Computers in the 1980 — trends in hardware technology; w INFORMATION PROCESSING 1974.

Podstawowymi składnikami czwartej generacji są półprzewodniki krzemowe — elementy bipolarne i unipolarne z izolowaną bramką (MOS — *metal-oxide semiconductor*). Inne elementy ciała stałego, pozostające jeszcze w laboratoriach, obejmują elementy z przesuwaniem magnetycznych ścian domenowych, elementy o sprzężeniu ładunkowym (CCD — *chargecoupled devices*) i kriogeniczne elementy ze złączem Josephsona (*oryogenic electron-tunneling devices*). Elementy te są atrakcyjne dzięki ich prostym strukturom i dużej gęstości upakowania, lecz oceniany dla nich podstawowy czas przełączania przy wykorzystaniu w układach logicznych jest większy niż wartość 0,02 nanosekundy, przyjmowanej jako wartość graniczna dla półprzewodników krzemowych. Stąd prognozowanie osiągów przyszłych komputerów w oparciu tylko o przewidywany postęp w dziedzinie półprzewodników nie powinno być obarczone błędami wynikającymi z przeoczenia obecnie znanych, szybszych elementów technicznych.

Bipolarne półprzewodnikowe układy logiczne

Najszybszymi półprzewodnikowymi układami przełączającymi są elementy logiczne ze sprzężeniem emiterowym (ECL — *emittercoupled logic*) wykorzystujące tranzystory bipolarne. Czas przełączania t_D bipolarnego elementu logicznego zależy od wewnętrznego opóźnienia t_{Di} tranzystorów bipolarnych i od stałych czasu związanych z geometrią układu, połączeniami i obciążeniem. To ostatnie może być zmieniane przez różnicowanie wartości przełączanych prądów (a stąd poboru mocy w układzie). Aby uniknąć złożonych obliczeń, można założyć, że $t_D \approx 3\ t_{Di}$. Wewnętrzne opóźnienie t_{Di} zależy głównie od czasu przelotu nośników mniejszościowych w obszarze bazy, $t_{Di} = KW_b^2$, gdzie W_b jest szerokością bazy, a stała K związana jest z ruchliwością nośnika w materiale dla danej temperatury. Dla temperatury pokojowej i dla ruchliwości nośnika 1000 $cm^2/V \cdot s$, $K = 3 \times 10^{-10}$. Stąd polepszenie szybkości elementu bipolarnego (tzn. zmniejszenie t_{Di}) związane jest bezpośrednio z możliwością wykonania bazy o bardzo małej szerokości. Przy szybkościach odpowiadających części nanosekundy wewnętrzne opóźnienie wykazuje tendencje do stawania się liniową funkcją szerokości bazy W_B i osiąga wartość graniczną 0,02 ns.

Konwencjonalna metoda przygotowywania masek wykonawczych dla produkcji tranzystorów i układów scalonych wykorzystuje techniki ortolitograficzne. Minimalna osiągalna szerokość bazy dla tej techniki jest około 1,5 mikrona (μm). Jednakże stosowana jest już nowa technologia produkcji wykorzystująca wiązkę elektronów lub jonów, która według oszacowań pozwala uzyskiwać graniczne szerokości bazy (wymiar liniowy razy 1,5) w zakresie 0,08 μm. Przewidywania przedstawione w tym artykule zakładają jednak, że praktycznie w 1990 roku szerokość bazy będzie 0,15 μm. W tabeli I przedstawiono przewidywane opóźnienie oraz inne właściwości bipolarnych elementów w okresie 1975—1990.

TABELA I. Właściwości bipolarnych układów logicznych

Rok	Minimalna szerokość bazy (μm)	Minimalne opóźnienie elementu t_D (ns)	Straty mocy (mW)	Maksymalne upakowanie elementów (elem./cm³)	Koszt elementu (w dolarach)
1975	0,66	0,4	55	55	0,30
1980	0,48	0,2	35	90	0,10
1985	0,25	0,1	25	125	0,05
1990	0,15	0,06	20	155	0,03

Układy logiczne z elementami unipolarnymi

Półprzewodnikowe elementy polowe z izolowaną bramką (MOS) różnią się od elementów bipolarnych kilkoma cechami: mają one prostszą strukturę, mniejszą powierzchnię i mniejszy pobór mocy. Jednakże opóźnienie dla elementów MOS jest prawie 20 razy większe, głównie w wyniku oddziaływania szkodliwych pojemności. Jedną z bardziej efektywnych technologii jest rozwiązanie „krzem na szafirze" w połączeniu z komplementarnymi układami logicznymi MOS (CMOS/SOS). Przewidywane opóźnienia układów oraz inne właściwości dla tych elementów przedstawiono w tabeli II.

TABELA II. Przewidywane właściwości układów CMOS/SOS

Rok	Opóźnienie elementu (ns)	Straty mocy		Maksymalne upakowanie	Koszt elementu (w dolarach)
		przełączanie (mW)	wyłączony (uW)	(elem./cm³)	
1975	1,0	5,6	10	1 100	0,02
1980	0,5	2,2	5	2 800	0,005
1985	0,28	1,0	2	6 000	0,003
1990	0,15	0,35	0,8	17 500	0,001

Tendencje rozwoju możliwości procesorów

Szybkość osiągana przez procesory zależy od rozwiązania systemowego i sprzętowego w zakresie architektury, od czasów wykonywania rozkazów, od szybkości układów podstawowych oraz od efektywności programów. Wzajemne oddziaływania wszystkich tych czynników są zbyt złożone, aby umożliwić przewidywania szybkości przetwarzania w przyszłości. Dla uproszczenia problemu można oszacować maksymalną wydajność dla określonego zestawu instrukcji (w pracy dla 30% mnożeń przy stałym przecinku i 70% dodawań przy stałym przecinku), przy uwzględnieniu przewidywanego czasu dostępu dla pamięci z dowolnym dostępem oraz spodziewanych ulepszeń „przezroczystości" pamięci, uzyskiwanych przez wstępne przeglądanie instrukcji i danych przed przetwarzaniem.

Przy stosowaniu tak uproszczonego modelu szybkość procesora S, w milionach instrukcji na sekundę (MIPS), jest określona przez $S = 1/t_o$, gdzie efektywny czas wykonywania instrukcji $t_o = 0{,}7\ t_{add} + 0{,}3\ t_{mult} + 2\ k_m t_{mc}$, czasy operacji arytmetycznych odpowiadają działaniom typu rejestr-rejestr, t_{mc} jest czasem dostępu do pamięci, a k_m przedstawia stopień przezroczystości pamięci. Przy wykorzystaniu tej zależności można prognozować osiągi dla poszczególnych rozwiązań architektonicznych. Założono przy tym, że wszystkie procesory działają na słowach 64-bitowych.

Rozwiązania jednoprocesorowe

Jednoprocesorowym rozwiązaniom odpowiadają takie procesory, w których występują pojedyncze strumienie instrukcji i pojedyncze strumienie danych, chociaż mogą one zawierać specjalizowane jednostki dla równoległych obliczeń i operacji wejścia-wyjścia. Prawie wszystkie współczesne komputery należą do tej klasy, włączając te procesory CDC 7600 oraz serii IBM 360 i 370.

Postępu w rozwoju architektury rozwiązań jednoprocesorowych można oczekiwać przez dołączanie specjalizowanych procesorów, szybkich arytmometrów, szybkich pamięci buforowych i rejestrów o organizacji stosowej, złożonych mechanizmów wstępnego przeglądania oraz rozwiązań architektonicznych pamięci zwiększających jej przezroczystość, takich jak rozszerzone tory przesyłania danych, moduły pamięci z przeplataniem i podobne.

Przewidywane możliwości dla rozwiązań systemów jednoprocesorowych przedstawiono w tabeli III.

Modyfikacją architektoniczną rozwiązania jednoprocesorowego jest oddziaływanie wielu instrukcji na jeden (lub kilka) strumieni i danych, przechodzących przez linie (*pipelines*) kolejno połączonych jednostek wykonawczych. Przykładami są rozwiązania CDC STAR-100, IBM 360/195 i TEXAS INSTRUMENTS ASC. Jeśli każdy z M procesorów ma w przybliżeniu taki sam czas wykonywania operacji t_s, i przetwarzany jest zestaw złożony z N słów danych, wówczas efektywny czas wykonania operacji dla

TABELA III. Możliwości rozwiązania jednoprocesorowego

Rok	Opóźnienie elementu (ns)	t_{add} (ns)	t_{mult} (ns)	t_{mc} · (ns)	k_m	Rozwiązanie jednoprocesorowe (MIPS)	Procesor typu linia (MIPS)
1975	0,4	7	95	20	0,25	23	120
1980	0,2	3,5	32	10	0,15	66	330
1985	0,1	1,2	16	5,6	0,125	140	715
1990	0,06	0,7	6	3	0,1	345	1740

jednego słowa wynosi $T_w = (T_o + Mt_s + (N-1)t_s)N$, gdzie T_o jest czasem przygotowania linii do pracy. Dla dużych wartości N, $T_w \approx t_s$. Wynikiem tego jest całkowita przezroczystość pamięci (tzn. $k_m = 0$). W tabeli III uwzględniono również przewidywane możliwości dla rozwiązania architektonicznego o organizacji liniowej.

Procesory siatkowe

Procesory siatkowe są systemami o pojedynczym strumieniu instrukcji i wielu strumieniach danych, dla których typowo występuje N identycznych elementów przetwarzających sterowanych przez ten sam strumień instrukcji, tzn. że wszystkie elementy przetwarzające działają zgodnie. Takim systemem jest procesor ILLIAC IV. Jeśli wszystkie N jednostek przetwarzających działa w tym samym czasie, wówczas uzyskuje się N-krotne zwiększenie szybkości w porównaniu do pojedynczego procesora. Zależy to jednak od typu wykonywanego zadania przy czym z reguły rzadko można wykorzystać pełne wykorzystanie elementów przetwarzających.

Ze względu na dużą liczbę elementów przetwarzających i możliwości znacznego zwiększenia szybkości dzięki architektonicznemu rozwiązaniu procesora siatkowego, poszczególne procesory mogą być proste i można oczekiwać, że nie będą miały własności wstępnego przeglądania (tj. $k_m = 1$). Pamięci związane z indywidualnymi procesorami odgrywają natomiast rolę szybkich buforów. Tabela IV przedstawia przewidywane szybkości procesorów siatkowych.

TABELA IV. Przewidywane wydajności procesora siatkowego

Rok	Opóźnienie elementu (ns)	t_{add} (ns)	t_{mult} (ns)	t_{mc} (ns)	k_m	Wydajność $N=1$ (MIPS)	Wydajność $N=64$ (MIPS)
1975	0,4	7,0	95	20,0	1	14	880
1980	0,2	3,5	50	10,0	1	26	1700
1985	0,1	1,8	25	5,6	1	50	3200
1990	0,06	0,9	12	3,0	1	98	6250

Inne rozwiązania architektoniczne

Istnieją również prognozy wydajności dla dwóch innych rozwiązań architektonicznych systemów komputerowych — konfiguracji wieloprocesorowej i asocjacyjnych procesorów siatkowych. Konfiguracja wieloprocesorowa stanowi system z wieloma strumieniami danych, w którym każdy z P procesorów jest właściwie złożonym procesorem pojedynczym. Procesory są wykorzystywane do rozwiązywania we współpracy jednego problemu, lub do pracy nad różnymi problemami, lecz przy sterowaniu przez system operacyjny, który jest umieszczony w jednym z tych procesorów. Z tego względu, oraz w wyniku istnienia potrzeby synchronizacji procesorów na różnych etapach współpracy przy wspólnej realizacji obliczeń, wzrost szybkości w porównaniu do pojedynczego procesora jest mniejszy od P. Przykładami współczesnych rozwiązań wieloprocesorowych są serie komputerów BURROUGHS 6700, UNIVAC 1100 i HONEYWELL 7000.

Asocjacyjne procesory siatkowe stanowią specjalny rodzaj procesorów siatkowych, w których N elementów przetwarzających jest kolejno, pojedynczo dołączanych do pamięci asocjacyjnej.

N słów w pamięci asocjacyjnej jest jednocześnie przetwarzane w m kolejnych krokach. Zwiększenie wydajności uzyskuje się wtedy gdy $N \gg m$ i operacje są proste (porównania logiczne a nie mnożenia). Dostępnym na rynku asocjacyjnym procesorem siatkowym jest STARAN IV firmy GOODYEAR. Postępy w rozwoju wydajności są warunkowane szybszymi układami logicznymi procesorów i w szczególności wzrostem szybkości pamięci asocjacyjnej.

Komputery w systemach sterowania i naprowadzania

Poprzednio omówione systemy komputerowe działają w ustalonych, dobrze przygotowanych i troskliwie kontrolowanych warunkach.

Istnieje jednak wiele zastosowań komputerów, gdy komputer musi pracować w warunkach otoczenia, które tylko częściowo można kontrolować, np. gdy komputer znajduje się w poruszającym się pojeździe lub co najmniej musi być łatwy do transportowania. Takie systemy nazywają się systemami sterowania i naprowadzania i zwykle narzucają specjalne wymagania na komputery. Przykładami takich wymagań mogą być: wytrzymałość na duże zmiany klimatyczne, wibracje mechaniczne i udary, promieniowanie jądrowe, małe wymiary, mały pobór mocy i bardzo duża niezawodność.

Komputery w systemach sterowania i naprowadzania można podzielić na naziemne, lotnicze i kosmiczne. W każdym z tych rodzajów stosowane są różne rozwiązania architektoniczne. Rozwiązania wieloprocesorowe są specjalnie atrakcyjne do stosowania w systemach sterowania i naprowadzania, ponieważ zapewniają „łagodną degradację", tzn. jeśli jeden z procesorów działa wadliwie, wówczas inne mogą podtrzymywać działanie systemu, chociaż z mniejszą efektywnością.

Wymagania dotyczące specjalnych wymiarów oraz odporności dla komputerów w systemach sterowania i naprowadzania kolidują z uzyskaniem tego samego poziomu osiągów jak dla komputerów ogólnego przeznaczenia. Jednak nowe osiągnięcia dotyczące układów logicznych, a w szczególności technika LSI, pozwalają na znaczne zwiększenie wydajności, polepszenie niezawodności i zmniejszenie wymiarów oraz ciężaru. Jednym z przykładów jest komputer *„All Aplication Digital Computer"* (AADC) opracowany przez marynarkę wojenną USA.

System jest zbudowany z modułów o szybkości od 2 do 3 MIPS, upakowanych w objętości mniejszej niż 750 cm³ i ważących mniej niż 1,5 kg.

Systemy pamięci

W obecnych pamięciach z dostępem swobodnym czasy dostępu są kilka rzędów wielkości i większe niż czasy opóźnień układów logicznych. Ponadto rosną one wraz ze wzrostem pojemności pamięci. Wynika stąd potrzeba stosowania różnych rozwiązań architektonicznych pamięć-system, aby usunąć niedopasowanie szybkości „logiki" i pamięci, takich jak układy hierarchiczne jednostek pamięci z szybkimi przesłaniami blokowymi, systemy pamięci z przeplataniem oraz rozszerzone tory dostępu.

Najbardziej obiecujący w pracach nad szybkimi pamięciami z dostępem swobodnym jest rozwój pamięci półprzewodnikowych, szczególnie pamięci buforowych. Zapowiada to zmniejszenie niedopasowania szybkości układów logicznych i pamięci. Ponieważ pobór mocy jest ważnym czynnikiem dla pamięci półprzewodnikowych, więc są raczej stosowane elementy CMOS/SOS, a nie bipolarne tranzystory. Tabela V przedstawia prognozy właściwości półprzewodnikowych pamięci buforowych.

TABELA V. Własności pamięci buforowych CMOS/SOS (256 k bitów)

Rok	Opóźnienie elementu t_D (ns)	Czas odczytu/zapisu $20t_D$ (ns)	Gęstość (bitów/cm³)	Moc na bit (μW)	Koszt na bit (w dolarach)
1975	1,0	20,0	450	24,0	0,007
1980	0,5	10,0	900	15,0	0,003
1985	0,28	5,6	1800	5,0	0,0018
1990	0,15	3,0	1900	0,5	0,0012

Nowe składniki, takie jak elementy z przesuwaniem magnetycznych ścian domenowych i elementy o sprzężeniu ładunkowym (CCD), będą wykorzystywane do konstruowania pamięci zastępujących obecne magnetyczne pamięci bębnowe i dyskowe. Spowoduje to znacznie ulepszenie wszystkich własności pamięci poczynając od szybkości do rozmiarów i niezawodności.

W dziedzinie pamięci masowych największe potencjalne możliwości stwarzają pamięci elektrooptyczne (10^{12} do 10^{13} bitów z dostępem do bloku od 5 do 10 mikrosekund).

Propozycje innowacji

Przewidywane tendencje rozwoju sprzętu pozwalają nie tylko na istotne ulepszenia takich tradycyjnie związanych ze sprzętem własności jak szybkość, czas dostępu pa-

mięci, pojemność pamięci, wymiary i niezawodność, lecz także istotnie wpływają na ułatwienia w wykorzystywaniu systemu komputerowego. Następujące dziedziny związane z funkcjonowaniem współczesnych systemów komputerowych, odpowiadają dziedzinom, w których innowacje są najniezbędniejsze, oraz na które przyszły rozwój techniczny sprzętu może wywierać największy wpływ.

● **Przygotowywanie, przekazywanie i konserwacja oprogramowania.** Sprzęt może być wykorzystywany do zapewnienia „programowej przezroczystości" zapewniającej uproszczenie w kontaktowaniu się programistów ze sprzętem. Systemy przetwarzania działające bezpośrednio w językach wyższego rzędu są ilustracją takiej możliwości.

● **Modularne, rozszerzalne systemy komputerowe i sieci komputerowe.** Modularność przy projektowaniu zapewni zestawienie efektywnych ze względu na koszty i problemowo zorientowanych konfiguracji systemowych, dostosowanych do określonych potrzeb obliczeniowych.

● **Wysoko rozwinięta komunikacja człowiek-maszyna.** Efektywne wzajemne oddziaływanie będzie rosnącym ze względu na swą wagę wymaganiem. Obecnie są rozwijane „inteligentne" terminale, lecz jest to hamowane przez koszty sprzętu. Zastosowanie naturalnego języka do komunikacji człowiek-maszyna może zapewnić bardziej elastyczną współpracę, lecz obecnie wymaga znacznego zakresu przetwarzania.

W latach osiemdziesiątych takie możliwości przetwarzania będą dostępne po realnym koszcie.

● **Sterowanie możliwością dostępu.** W systemach zdalnie dostępnych z podziałem zasobów i wieloma użytkownikami niezbędne jest zagwarantowanie każdemu użytkownikowi nienaruszalności jego danych i programów oraz możliwości pełnej kontroli dostępu do jego zasobów. Obecne systemy sterowania dostępem, oparte na rozwiązaniach programowych, nie mogą być zrealizowane w stopniu dostatecznie efektywnym.

● **Żywotność systemu.** Powstawanie dużych systemów przetwarzania danych, szczególnie systemów sterowania i naprowadzania, stwarza postulat, aby systemy praktycznie nigdy nie przerywały swej pracy, a dane nigdy nie były tracone. Przyszły sprzęt o niskich kosztach będzie pozwalał na zaprojektowanie systemów niewrażliwych na uszkodzenia oraz systemów z samodiagnozą i z możliwością samoistnej naprawy. Systemy o 10-letniej wartości średniego czasu pomiędzy uszkodzeniami, takie jak np. JPL-STAR, staną się praktycznie możliwe nie tylko w systemach międzyplanetarnych pojazdów kosmicznych.

Do innych innowacji, jakie mogą być realizowane w oparciu o przyszły rozwój sprzętu komputerowego, należy rozdział funkcji pomiędzy sprzęt, oprogramowanie i środki programowo-sprzętowe *(firmware)* celem optymalizacji kosztu całego systemu, zmniejszenia złożoności oprogramowania i zapewnienia większej elastyczności oraz zdolności adaptacyjnej systemu.

Uwagi końcowe

Powyżej przedstawiony skrótowo potencjalny rozwój osiągów komputerów i systemów pamięci stanowi stosunkowo małą część rozważań związanych z projektowaniem i stosowaniem komputerów. Ze względu na brak miejsca nie rozpatrzono wszystkich elementów prognoz.

Czy przedstawione prognozy sprawdzą się? Martino pokazał, że miarą wartości prognozy nie jest to, czy ona się sprawdzi, lecz raczej to czy zapewnia ona informację dla podjęcia właściwych decyzji.

Tłum. i oprac. **A. Kojemski**

PRZEGLĄD WYDAWNICTW

Bibliografia wydawnictw polskich z dziedziny informatyki

● **Kierowanie produkcją samochodów — PLUCIŃSKI S.** Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1975, s. 38, cena zł 10.— II Telewizyjny Kurs Informatyki. Poradnik Zawodowy D/22, 23.

Wprowadzenie. Charakterystyka Fabryki Samochodów Ciężarowych w Starachowicach. Obiektowy system informatyczny. Organizacja zarządzania produkcją w FSC. Odcinkowy system informatyczny kierowania produkcją.

● **Podstawowe problemy przygotowania użytkowników systemów informatycznych — TRYBUSKI J.** Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1975, ss. 30, cena zł 10.— II Telewizyjny Kurs Informatyki. Poradnik Zawodowy E/33, 34.

Wprowadzenie. Elementy przygotowania systemu. Projekt systemu. Uczestnicy procesu przygotowania systemu — Kwalifikacje. Kierownictwo — Koordynacja.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Zadania kontrolne m.c. R-30.21.C40013-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 31.

Przewodnik po zadaniach kontrolnych. Wykaz i opis zadań kontrolnych. Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Testy urządzeń zewnętrznych m.c. R-30. Testy autonomiczne. 21.C40004-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 58.

Testy: Konsoli monitora (KNTP), drukarki wierszowej KNTPI, czytnika taśmy papierowej (KNTP), dziurkarki taśmy papierowej (KNTP), czytnika kart (KNTP), dziurkarki kart (KNTP).

Materiały przeznaczone są dla operatorów i programistów.

● **Sterowanie numeryczne obrabiarek. Poradnik — SHAH R.** Tłum. wyd. ang. z 1971 r. WNT, Warszawa 1975, ss. 146, cena zł 24.—

Planowanie. Uzasadnienie ekonomiczne. Kryteria wyboru. Wskazówki dotyczące wykorzystania obrabiarek Nr 44. Określenia podstawowych pojęć. Programowanie ręczne. Programowanie automatyczne. APT.EXAPT 1, EXAPT 2.

Poradnik przeznaczony jest dla inżynierów i techników technologów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Testy urządzeń zewnętrznych m.c. R-30. Test — sekcje DMES. C40005-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 19.

Testy: funkcjonalno-kontrolny konsoli monitora z drukarki wierszowej. Test trudnego reżimu pracy drukarki wierszowej.

Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Testy pamięci zewnętrznych m.c. R-30. Test — sekcje DMES, Z.2.C40008-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 38.

Test funkcjonalno-kontrolny jednostek taśm magnetycznych. Test zużycia i wymienności taśm magnetycznych.

Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Programy organizacyjne m.c. R-30. Opis programu DMES Z.1. C40015-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 58.

Funkcja programu. Wymagania programu. Przedmiot sprawdzania. Instrukcje operatorskie. Komunikaty. Sygnalizacja błędów. Symbol programu. Dodatki.

Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Informacje wstępne o testach m.c. R-30. C40001-1.** Wyd. MERA-ELWRO. Wrocław 1975, ss. 22.

Podział, budowa, dokumentacja i biblioteka testów. Dodatki. Wykaz podręczników. Program ładujący z taśmy magnetycznej.

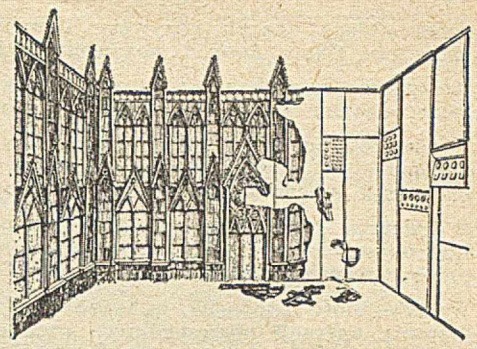
Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. DOS/JS. Assembler F.C12102-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 7.

Translatory Assembler E i Assembler F. Różnice w języku dla translatorów Assembler E i Assembler F. Dodatki: Komunikaty diagnostyczne, komunikaty o błędach translacji na SYSLOG.

Materiały przeznaczone są dla operatorów i programistów.

dokończenie na str. 47



JERZY CHMURZYŃSKI
Wydział Cybernetyki WAT
Warszawa

Laboratorium programowania współbieżnego (1)

Zagadnienie przetwarzania współbieżnego dawno już przestało być domeną zainteresowań wyłącznie teoretycznych, stając się również techniką coraz powszechniej stosowaną przez programistów i projektantów systemów komputerowych. Głównym powodem bujnego rozkwitu techniki przetwarzania współbieżnego jest możliwość znacznego przyśpieszenia przetwarzania oraz polepszenie współczynników wykorzystania instalacji komputerowej.

Jedną z trudności spotykanych podczas stosowania tej techniki jest brak odpowiedniego języka programowania [8]. Dotychczas zaproponowana została spora liczba konstrukcji językowych, przydatnych do opisywania problemów przetwarzania współbieżnego [3, 7, 8].

Niezależnie od prac mających na celu zbudowanie najbardziej odpowiedniego języka istnieje możliwość praktycznego wykorzystania wspomnianych wyżej konstrukcji. Konstrukcje te, jakkolwiek nie zostały nigdzie zebrane łącznie i wykorzystane w języku programowania, to jednak same w sobie stanowią pewien **podzbiór języka abstrakcyjnego**, przydatnego do modelowania, opisywania i rozwiązywania problemów przetwarzania współbieżnego. Włączenie podzbioru języka abstrakcyjnego do konkretnego języka programowania np. PASCALA, dostarcza **abstrakcyjnego języka programowania.**

Posługiwanie się językiem abstrakcyjnym programowania w znacznym stopniu upraszcza projektowanie i konstrukcję złożonych systemów komputerowych.

Mgr inż. Jerzy CHMURZYŃSKI (1947) ukończył Wydział Cybernetyki Wojskowej Akademii Technicznej, specjalność maszyny matematyczne (1972). Obecnie jest starszym asystentem w Katedrze Maszyn Matematycznych WAT. Zajmuje się systemami operacyjnymi i inżynierią oprogramowania komputerów. Uczestniczy w pracach nad systemami pracującymi w czasie rzeczywistym.

Język taki może być również wykorzystywany do nauczania zasad budowy złożonych systemów oprogramowania. W Katedrze Maszyn Matematycznych Wydziału Cybernetyki WAT wprowadzone zostały zajęcia laboratoryjne, podczas których studenci mają możliwość praktycznego zapoznawania się z nowymi technikami programowania. Laboratorium Programowania Współbieżnego (LPW), uruchomione na komputerze ODRA-1305 umożliwia programowanie wieloprogramowe i wieloprocesowe. Zestaw programów użytkowych, stanowiących wyposażenie LPW, dostarcza możliwości demonstrowania i badania szerokiej klasy problemów programowania współbieżnego, jak inicjowanie i kończenie procesów, wzajemne wykluczanie procesów, synchronizacja wzajemna procesów na bazie semaforów, synchronizacja czasowa, problemy blokowania się procesów, realizacja programowa konstrukcji językowych typu „region krytyczny", „warunkowy region krytyczny" *„co-begin-coend"*, *„cycle-end"*, współdziałanie procesów za pomocą buforów komunikacyjnych i inne. W artykule omówiony zostanie sposób zinterpretowania niektórych z tych pojęć na grucie LPW i sposobu ich wykorzystywania w programowaniu.

Artykuł jest jednocześnie pierwszą częścią opracowania, w którym przedstawione zostanie LPW.

Przedstawione będą tu główne abstracyjne konstrukcje językowe, wykorzystywane w laboratorium do synchronizacji procesów współbieżnych.

W drugiej części omówiona będzie dokładniej budowa laboratorium oraz poszczególne ćwiczenia, które mogą być realizowane za jego pomocą. Trzecia i ostatnia część dotyczyć będzie monitora STER, stanowiącego podstawowy element laboratorium.

Aparat synchronizacji procesów współbieżnych

LWP umożliwia inicjowanie i kończenie procesów oraz dostarcza możliwość budowania złożonych aparatów synchronizacji procesów.

Pojęcie **procesu** jest pojęciem kluczowym we współczesnych technikach programowania. Projektowanie i konstrukcja złożonych systemów komputerowych znacznie się upraszcza, jeżeli potrafimy problem we właściwy sposób zdekomponować. Procesy, jako podstawowe jednostki dekompozycji złożonych systemów umożliwiają przedstawienie dynamicznej struktury tych systemów [1, 2, 5, 11]. **Proces** jest to sekwencja operacji odwzorowujących skoń-

czone zbiory danych nazywane **danymi wejściowymi** w skończone zbiory danych, nazywane **wyjściowymi** [7].

Komputerowa realizacja operacji poszczególnych procesów może odbywać się na dwa różne sposoby:

- pierwsza operacja dowolnego procesu rozpoczyna się po zakończeniu ostatniej operacji innego procesu; jest to przypadek **przetwarzania sekwencyjnego;**
- operacje dowolnego procesu są rozpoczynane przed zakończeniem ostatnich operacji innych procesów; jest to przypadek **przetwarzania współbieżnego.**

Współbieżność procesów osiągana jest albo drogą **przeplatania operacji** poszczególnych procesów w czasie albo też drogą **równoczesnego wykonywania operacji** różnych procesów, w zależności od liczby procesorów fizycznych, których używamy do realizacji procesów. Możliwe jest oczywiście osiąganie współbieżności poprzez połączenie obu powyższych sposobów.

Procesy współbieżne charakteryzują się różnym stopniem wzajemnego oddziaływania. Procesy wykorzystujące rozłączne zbiory danych wejściowych i wyjściowych nazywać będziemy procesami **rozłącznymi** [4, 7, 10].

Częstszym i o wiele kłopotliwszym przypadkiem procesów współbieżnych spotykanych w systemach komputerowych są procesy **oddziałujące** na siebie. Głównymi powodami oddziaływania procesów jest konieczność dzielenia ograniczonej puli resursów systemu oraz wykorzystywanie do celów komunikacji między procesami zmiennych globalnych. W takich sytuacjach ważne jest zazwyczaj, aby w danej chwili tylko jeden proces (lub ściśle określona grupa procesów) wykorzystywał resurs lub korzystał z prawa dostępu do zmiennych globalnych.

Proces, wykorzystujący w sposób wyłączny resurs realizuje sekwencję operacji, którą nazywa się **regionem krytycznym** [3, 4, 7]. Wejście procesu do regionu krytycznego pociąga za sobą automatycznie uniemożliwienie pozostałym procesom wejścia do tego regionu. Po opuszczeniu przez proces regionu krytycznego resurs jest zwalniany i może być z kolei wykorzystany przez inny proces; proces ten powinien w tym celu wejść do regionu krytycznego, związanego ze zwolnionym resursem.

Tak więc region krytyczny ustanawia stopień szeregowania operacji współbieżnych niezbędny do zapewnienia, aby w danej chwili co najwyżej jeden proces wykorzystywał resurs lub zmienną globalną.

W celu określania regionów krytycznych w programach używana jest następująca konstrukcja językowa:

var V: **shared** T ;

Za pomocą tej notacji deklarowana jest zmienna globalna *V* typu *T* [7]. Procesy współbieżne mogą odwoływać się do zmiennych globalnych jedynie wewnątrz regionu krytycznego, który definiujemy następującą konstrukcją językową:

region V **do** S ;

Notacja ta umożliwia realizację operacji *S* na zmiennej globalnej *V* w sposób wyłączny, tzn. z wykluczeniem dostępu innych procesów do tej zmiennej. Zmienna *V* może być zmienną globalną, wykorzystywaną przez procesy współbieżne do celów komunikacji, albo też może to być zmienna globalna opisująca stan resursu fizycznego. Przedstawiona powyżej notacja regionu krytycznego umożliwia więc realizację **wzajemnego wykluczania** procesów w obu tych przypadkach.

Częstym przypadkiem oddziaływania procesów współbieżnych jest ich **współdziałanie** na drodze wymiany sygnałów lub meldunków. Proces może więc nadawać sygnał (meldunek) do innego procesu, czekać na przekazanie mu lub odbierać sygnał (meldunek), nadawany przez inny proces. Współdziałanie takie jest więc formą uzależniania jednych procesów od drugich. Współdziałanie procesów powinno odbywać się według określonych, niezmiennych podczas wymiany sygnałów (meldunków) zasad.

W związku z powyższym współdziałanie realizowane jest najczęściej z wykorzystaniem standardowych obiektów pośredniczących wymianie, do których sygnały (meldunki) mogą być nadawane i z których mogą być pobierane. Ponieważ zarówno nadawanie jak i odbieranie sygnałów do/z obiektów pośredniczących odbywa się za pomocą standardowych procedur, to możliwym się staje określenie i przestrzeganie zasad wymiany. Wymianę sygnałów dogodnie jest realizować wykorzystując w tym celu semafory [2, 3, 7], deklarowane w programie w sposób następujący:

var S: **semaphore** ;

Z semaforem związane są dwie standardowe operacje nadawania i pobierania sygnałów. Operacja *signal (S)* umożliwia procesowi przekazywanie jednego sygnału do semafora *S*, natomiast operacja *wait (S)* umożliwia pobranie jednego sygnału z semafora *S*; w przypadku, gdy w semaforze brakuje sygnałów, proces wykonujący operację *wait (S)* zostaje opóźniony do czasu aż inny proces, za pomocą operacji *signal (S)* przekaże sygnał do semafora *S*.

Obiektem pośredniczącym w wymianie meldunków między procesami jest bufor komunikacyjny [7]. Za pomocą notacji:

var B: **buffer** c **of** T ;

deklarujemy bufor *B* o pojemności *c*, przechowujący meldunki typu *T*. Pojemność bufora mierzona jest ilością meldunków. Nadawanie meldunku *M* do bufora *B* możliwe jest za pomocą standardowej operacji *send (M, B)*. W przypadku gdy bufor jest przepełniony, tzn. gdy ilość meldunków w nim zgromadzonych równa jest jego pojemności, proces nadający jest opóźniany do momentu pojawienia się wolnego miejsca w buforze.

Proces może pobrać meldunek *M* z bufora *B* wykorzystując operację *receive (M, B)*. Jeżeli bufor jest aktualnie pusty, tzn. żaden meldunek nie został do bufora nadany, wtedy proces pobierający meldunek zostaje opóźniony do czasu pojawienia się meldunku w buforze.

Przedstawione powyżej mechanizmy synchronizacji umożliwiają budowanie założonych systemów procesów współbieżnych. Prezentowane konstrukcje językowe zapewniają jednocześnie prostotę koncepcyjną budowanych systemów oraz, co za tym idzie, w znacznym stopniu chronią przed popełnianiem błędów. Problematyka poprawności budowy złożonych systemów procesów współbieżnych obejmuje również tzw. blokadę procesów [5, 9]. Procesy, które uzyskały dostęp do resursu mogą próbować uzyskać dodatkowo dostęp do drugiego i dalszych resursów. **Blokada** procesów jest to taki stan systemu, w którym każdy z kilku procesów, dysponując częścią resursów systemowych jest opóźniany do czasu otrzymania kolejnych resursów, przydzielonych już do innych procesów.

Zablokowane procesy będą opóźniane tak długo, aż jakaś specjalna, zewnętrzna akcja systemu operacyjnego lub operatora nie spowoduje usunięcia tego opóźnienia.

Uwagi końcowe

Problem posiadania odpowiedniego języka opisu złożonych systemów komputerowych jest problemem dużej wagi, często decydującym o powodzeniu prac projektowych.

W przypadku projektowania jakiegokolwiek programu, a w szczególności programu bardzo rozbudowanego, dla uproszczenia i ułatwienia sobie tego zadania potrzebujemy pewnego narzędzia abstrakcji. Zakodowujemy więc program za pomocą abstrakcyjnych konstrukcji językowych i procedur, które funkcjonować będą być może jedynie w czasie pisania programu. Podobnie przedstawia się sytuacja z danymi dla programu, które również opisujemy za pomocą pewnych abstrakcyjnych pojęć, jak: rekordy, tablice, sekwencyjne struktury danych itp. Możemy przy tym założyć, że taki abstrakcyjny język nie będzie kompilowany automatycznie i przyjmiemy, że translacja i implementacja tak zaprojektowanego języka będzie odbywała się w sposób manualny. Będzie to trwało tak długo, dopóki wymagania stawiane językowi projektowania nie będą wyraźnie sprecyzowane, albo też kiedy potrafimy stworzyć taki język, który zapewni możliwość prostego i łatwego definiowania struktur danych konstrukcji językowych i procedur wymaganych przez użytkownika.

Przedstawione w artykule konstrukcje językowe są przykładem konstrukcji abstrakcyjnych.

„Wprowadzenie" ich do istniejącego języka programowania, np. języka PASCAL dostarcza w rezultacie pewnego abstrakcyjnego języka projektowania złożonych systemów komputerowych.

Z dotychczasowych doświadczeń zebranych w Katedrze Maszyn Matematycznych WAT wynika, że język taki jest bardzo dogodnym narzędziem projektowania, a ponadto dobrze nadaje się do nauczania zasad budowy oprogramowania złożonych systemów komputerowych.

LITERATURA

[1] BETOURNE C., FERRIE J., KAISER C., KRAKOWIAK S., MOSSIERE J. — System design and implementation using parallel processes. ACM SIGPLAN NOTICES vol. 7, nr 11 (1972).

[2] DIJKSTRA E. W. — The structure of the „THE" multiprogramming system. Comm. ACM 11,5 (1968).

[3] DIJKSTRA E. W. — Cooperating sequential processes. (w Programming Languages, Academic Press N. Y. 1968).

[4] GILBERT P., CHANDLER W. J. — Interference between communicating parallel processes. Comm. ACM 8,6 (1972).

[5] HABERMANN A. N. — On the harmonious cooperation of abstract machines. Mathematics Department, Technological University, Eindhoven, (1967).

[6] HANSEN P. B. — The nucleos of a multiprogramming system. Comm. ACM 10,5 (1967).

[7] HANSEN P. B. — Operating system principles. Prentice — Hall, Englewood Cliffs, New Jersey (1973).

[8] HOARE C. A. R. — Towards a theory of parallel programming. (w Operating Systems Techniques, Academic Press, New York 1973).

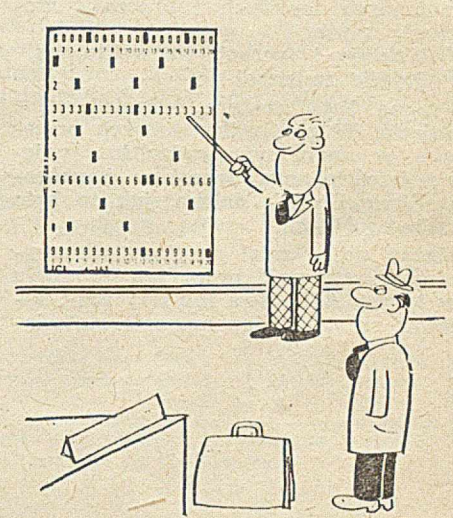
[9] HOLT R. C. — On deadlock in computer systems. Technical Report CSRG-6 (1971).

[10] HORNING J. J. RANDEL B. — Process structuring Technical Report CSRG-15 (1972).

[11] TURSKI W. M. — SODA — A dual activity operating system. Computer J. 11,2 (1968).

KADRY INFORMATYKI

Nauczanie matematyki na poziomie podstawowym przy użyciu języka APL



Artykuł ten jest krótkim opisem pierwszej części eksperymentalnego nauczania matematyki przy użyciu języka APL [1] w kilku warszawskich szkołach podstawowych.

Cztery lata temu miałem przyjemność prowadzić wykład z APL oraz jednocześnie z algebry liniowej dla studentów sekcji nauczycielskiej Wydziału Matematyki UW. Ponieważ APL został napisany na bazie notacji i formalizmu algebry liniowej, bardzo szybko nabrałem przekonania, że jest to język bardzo użyteczny do nauczania matematyki, nawet na poziomie podstawowym.

Za cztery lata wchodzi w życie nowy system szkolnictwa oparty na ogólnokształcącej szkole dziesięcioletniej. Jednym z przedmiotów w tej szkole będzie matematyka zintegrowana z informatyką. Naturalny jest więc problem: jaki język programowania wybrać, żeby kształcić matematycznie i wprowadzić podstawowe metody informatyczne, nie wprowadzając jednocześnie zbędnego na tym poziomie formalizmu związanego z większością języków programowania. Wymienione wymagania mogą być spełnione wyłącznie przez język konwersacyjny, np.: APL.

Celem eksperymentu jest próba takiego nauczania informatyki w szkole, aby było to jednocześnie związane z kształceniem matematycznym. Chciałbym tutaj wyraźnie podkreślić, że w proponowanym projekcie APL nie jest i nie ma być narzędziem do nauczania informatyki, ale narzędziem do ogólnego matematycznego kształcenia, używając komputerów jako nowoczesnych maszyn do liczenia.

Spośród istniejących rozwiązań jedynie APL daje możliwość takiego postawienia problemu. Obok faktu, że jest to język konwersacyjny, równie ważny jest fakt, że APL jest semantycznie bardzo zbliżony do algebry liniowej dobrze przecież znanej wszystkim nauczycielom matematyki.

W pierwszym semestrze roku 1974/75 wybrana grupa studentów z II roku sekcji nauczycielskiej na Wydziale Matematyki UW poznała język APL. Oprócz wykładów brali oni udział w seminarium, na którym zostały omówione zastosowania tego języka do nauczania matematyki. W następnym semestrze rozpoczęliśmy naukę matematyki przy użyciu APL.

Eksperyment obejmował uczniów z ośmiu klas VI w warszawskich szkołach podstawowych wybranych do tego celu raczej losowo. W sumie nauczaliśmy około 250 uczniów przez trzy miesiące.

W celu uniknięcia jednostronnego spojrzenia na przebieg eksperymentu, nauczanie pozostawało w rękach studentów, chociaż było dla mnie zupełnie realne osobiste prowadzenie nauczania w tych wszystkich klasach.

[1] Należy wyjaśnić, że w ostatnich latach daje się zauważyć duży wzrost zainteresowania językiem programowania APL, opracowanym i rozwiniętym przez firmę IBM. Teoretyczne podstawy tego języka zostały sformułowane w książce K. E. Iversona „A Programming Language" wydanej w r. 1962 przez J. Wiley Inc. Nazwa języka pochodzi od pierwszych liter tytułu tej książki. APL jest językiem konwersacyjnym opartym na algebrze liniowej. Oprócz tego zawiera cały szereg operatorów użytecznych w problemach związanych z pozostałymi działami matematyki. Wbrew pozorom nie jest to język ukierunkowany do obliczeń czysto matematycznych lub inżynierskich. Pełniejszy opis tego języka, jak również jego zastosowań, zostanie podany w następnych numerach „INFORMATYKI".

Dzięki temu, że nauczanie prowadzili studenci mogłem śledzić przebieg eksperymentu w taki sposób, w jaki ewentualnie będzie wyglądało nauczanie matematyki przy użyciu APL w szkole: nauczyciele matematyki będą się uczyć APL, a następnie będą nauczać matematyki używając tego języka lub innego języka programowania, w który w przyszłości rozwinie się APL.

Nauczaniem objęty były wszystkie dzieci w danej klasie, bez względu na ich poziom, gdyż był to jedyny realny sposób na zidentyfikowanie tych części naszego programu nauczania, które należało zmienić, poprawić lub usunąć.
Program nauczania obejmował następujące pojęcia matematyczne:
- Oś liczbowa
- Operacje arytmetyczne
- Wartość bezwzględna, funkcja signum
- Ułamki, ułamki okresowe
- Porządki liniowe, nierówności
- Wyrażenia algebraiczne
- Wprowadzenie do logiki: wyrażenia logiczne, operacje logiczne
- Wprowadzenie do rachunku prawdopodobieństwa: permutacje, kombinacje, zdarzenia i ich prawdopodobieństwa, drzewa i język grafów dla zmiennych losowych i procesów stochastycznych
- Wektory i operacje na wektorach
- Pojęcie zmiennej.

Podana lista zawiera jedynie ogólne pojęcia. W przypadku dobrych postępów w nauce wprowadzaliśmy pojęcia bardziej zaawansowane i próbowaliśmy iść tak daleko, jak to było tylko możliwe.

Każdy uczeń miał w ciągu eksperymentu możliwość pracy na terminalu w języku APL, dwa lub trzy razy, zależnie od stopnia opanowania materiału. Zajęcia te odbywały się dzięki uprzejmości IBM SUPPORT CENTER w Warszawie. Taka liczba godzin pracy każdego ucznia na terminalu APL była dla potrzeb eksperymentu wystarczająca, gdyż na tym poziomie wystarcza, aby prowadzący lekcje dobrze znał APL, a praca na terminalu służy raczej do pokazowych ćwiczeń. Przebieg eksperymentu był śledzony i dyskutowany na specjalnym seminarium. Pozwalało to również na wymianę doświadczeń między prowadzącymi poszczególne klasy, oraz na omawianie przyszłych tematów nauczania. Postępy uczniów były badane za pomocą testów, prac klasowych i powtórzeń.

Eksperyment wykazał użyteczność APL do matematycznego kształcenia. Pełny opis wyników eksperymentu będzie zawarty w dwóch podręcznikach: jednym dla uczniów, drugim dla nauczycieli. Pierwszy będzie zawierał kurs matematyki przy użyciu APL, a drugi będzie komentarzem do tego kursu.

Chciałbym tutaj zwrócić uwagę na dwa ogólne problemy związane z nauczaniem matematyki przy użyciu języka programowania.
Pierwszy wiąże się z faktem, że dla matematyków i nie tylko dla nich istnieje wyraźna bariera między językiem programowania a językiem matematycznym. Ponieważ APL czerpie wiele semantycznych konstrukcji prosto z matematyki, staje się bardzo bliski zwykłej algebraicznej notacji i dlatego ta trudność jest znacznie zredukowana. Użycie APL do nauczania pozwala przejść tę barierę chociaż powstają pewne problemy dydaktyczne.

Drugi problem jest związany z tłumaczeniem języka programowania z jednego języka na drugi, w tym przypadku z angielskiego na polski. Takie tłumaczenie wymaga bardzo dokładnego i starannego opisu użytego słownictwa i notacji. Przykładem może być pojęcie zmiennej, które w języku polskim sprawia dużo kłopotów, nie znanych w krajach z językiem angielskim.

Równie ważny jest fakt użycia prawdziwego „dorosłego" komputera, a nie czegoś co by tylko symulowało. Podstawowe znaczenie ma bowiem fakt, że uczniowie czują, że problemy które oni rozwiązują używając komputera mogą być w ten sam sposób rozwiązywane przez profesjonalistów.

Użycie komputera powoduje też, że dla uczniów matematyka staje się bardzo realna, niemal odczuwalna i to zarówno jej część teoretyczna, jak i praktyczna, zapewniona przez działanie komputera. W tym miejscu fakt, że APL jest językiem konwersacyjnym ma szczególnie mocną wymowę. Informatyka wywarła duży wpływ na kształcenie matematyczne. W tej chwili trudno jest przewidzieć jak duży wpływ na nauczanie matematyki będzie miało użycie komputerów w szkole. Prace badawcze na tym polu są dotąd niezwykle rzadkie, języki programowania w większości nie dają się zastosować do tego celu, eksperymenty dydaktyczne są niezwykle trudne. Są one jednak konieczne do rozwoju nowoczesnych metod nauczania matematyki w naszych szkołach.

Nauczanie matematyki powinno być w przyszłości mniej czasochłonne i matematyka powinna być bardziej użyteczna, dlatego jako wniosek z opisanego powyżej eksperymentu pozwolę sobie wyrazić przekonanie, że właśnie język APL może być znakomicie wykorzystany do tych potrzeb.

Andrzej J. K. Plewicki

ETO W ZARZĄDZANIU

Organizacyjne problemy zastosowania ETO w zarządzaniu

Dotychczasowe publikacje na temat zastosowań komputerów w zarządzaniu bardzo jednostronnie naświetlały tę problematykę. Dominował w nich zdecydowanie technicystyczny punkt widzenia.
Rozważania ograniczały się do technicznych problemów związanych z możliwościami poszczególnych typów komputerów oraz opisów procesu projektowania systemów informatycznych.
Wyglądało to tak, jak gdyby organizacja była wyłącznie złożonym systemem informacyjno-decyzyjnym, którego sprawne funkcjonowanie zależy od szybkości i liczby przetworzonych w nim informacji. Można było powiedzieć, że informatycy traktowali organizację na wzór i podobieństwo komputera. Takie podejście musiało w końcu doprowadzić do konfliktów i niepowodzeń.
Właśnie pierwsze niepowodzenia w zastosowaniu komputerów w zarządzaniu spowodowały, że dostrzeżono błędy w dotychczasowym podejściu.
Fascynacja techniką zostaje więc stopniowo zastępowana kompleksowym podejściem, w którym poświęca się równie

dużo uwagi nie tylko problemom technicznym budowania systemów informatycznych, ale również ich społecznym, psychologicznym i organizacyjnym uwarunkowaniom.
Okazało się, że nie wystarczą tylko ogromne możliwości komputera, które też mają swoje ograniczenia, potrzebna jest jeszcze społeczna akceptacja zmian i zapotrzebowanie na te zmiany w samej organizacji, aby zastosowanie komputera stało się w pełni efektywne.
Spójrzmy więc na interesujący nas problem właśnie od strony zmian w organizacji, jakie wywoływane są zastosowaniem komputerów. W tym przypadku będą nas interesowały przede wszystkim zmiany jakie dokonują się w systemie społecznym organizacji.

Innowacja w organizacji

Z punktu widzenia prowadzonych tutaj rozważań wdrożenie systemu elektronicznego przetwarzania danych w zarządzaniu jest traktowane przede wszystkim jako istotna zmiana w dotychczasowych zasadach i formach funkcjonowania organizacji.
Zmiana ta ma charakter innowacji i pociąga za sobą określone konsekwencje organizacyjne, społeczne i psychologiczne dla całej organizacji. H. March i A. Simon definiują innowację w organizacji jako „zmianę wymagającą znalezienia i oceny nowych programów działania, które dotychczas nie wchodziły do repertuaru organizacji i nie mogą być zrealizowane poprzez zaplanowane metody i przesunięcia" [1]. Wynika stąd, że wprowadzanie komputera do organizacji musi być rozpatrywane w kontekście daleko idących zmian w jej systemie społecznym. Problematyka innowacji została już stosunkowo dokładnie opisana w odniesieniu do innowacji technicznych w organizacjach produkcyjnych. Rozpoznane zostały podstawowe uwarunkowania oraz charakterystyczne zjawiska organizacyjne, społeczne i psychologiczne towarzyszące tego typu innowacjom.
Stosunkowo mało wiemy natomiast o problematyce innowacji w sferze zarządzania, którą jest zastosowanie komputera do przetwarzania danych. Wydaje się jednak, że wiele wniosków opartych na badaniach innowacji technicznych można by odnieść również do zagadnień innowacji organizacyjnych, wiele jednak problemów jest specyficznych i wynika z innego charakteru procesów.

Oczywiście opór przeciwko zmianie kształtuje się różnie w poszczególnych momentach jej realizacji. Dla celów analitycznych możemy wyróżnić trzy takie etapy:
- prac badawczych i szczegółowego rozpoznania warunków zmiany. W takim etapie zainteresowani znają cel zmiany i jej podstawowe warunki. Nie wiedzą natomiast na czym zmiana ma polegać
- prac przygotowawczych i wdrożeniowych, w którym zainteresowani znają dokładnie projekt zmiany, ale nie wiedzą co ona oznacza w praktyce (nie wypróbowali jej)
- zbierania doświadczeń praktycznych, w którym zainteresowani mają pełną informację na temat zmiany.

W dalszym ciągu przedstawimy główne przejawy oporu przeciwko zmianie, jaką jest wprowadzenie do systemu zarządzania komputera w rozbiciu na wyróżnione wyżej trzy fazy realizacji.
Ogólną dyrektywą praktyczną w zakresie postępowania mającego na celu zmniejszenie oporu przeciwko innowacji, w każdej z faz jej realizacji, jest właściwe przygotowanie systemu społecznego organizacji do przyjęcia zmiany.
Niezależnie od spełnienia szeregu wymogów technicznych i organizacyjnych adaptujących zarówno samą innowację jak i system społeczny organizacji i wynikających z istoty usprawnienia, można przedstawić szereg propozycji poczynań zmierzających do przygotowania i przekonania poszczególnych uczestników organizacji o sensowności i celowości proponowanych zmian.
Chodzi tutaj przede wszystkim o:

— właściwe przedstawienie podstawowych celów zmiany i uzyskanie ich aprobaty ze strony zainteresowanych członków organizacji

— uświadomienie wszystkim członkom organizacji, a w szczególności jej personelowi kierowniczemu możliwych konsekwencji organizacyjnych, społecznych i psychologicznych wprowadzonej innowacji oraz przedstawienie sposobów przeciwdziałania powstającym z tego powodu zjawiskom negatywnym

— odpowiednie przygotowanie i przeprowadzenie przeszkolenia w zakresie istoty zmiany oraz jej konsekwencji wszystkich członków organizacji, którzy mają wpływ na

TABELA I.

$$\text{Uczucie} = f\left[\frac{\text{Lękliwe nastawienie wobec każdej zmiany}}{\text{Osobiste poczucie bezpieczeństwa}}\right] = \left[\frac{(\text{Nie obalone określone obawy}) + (\text{Konflikty z wierzeniami kulturowymi}) + (\text{Przyszłe zdarzenia krzywdzące i nastawiające przeciwko zmianie}) + (\text{Irytacja wywołana sposobem zmiany})}{(\text{Zaufanie do kierownictwa zw, zawod i grupy pracowniczej})\left[(\text{Niewzruszone pełne nadziei oczekiwania}) + (\text{Przyszłe zdarzenia krzywdzące nastawiające przychylnie do zmiany}) + (\text{Zadowolenie ze sposobu zmiany})\right]}\right]$$

(wg A.S. Judson, A. Managers Guide to Marking Changes, London 1966 cyt. za 2)

Podstawowym zagadnieniem, jakie wiąże się z wprowadzeniem innowacji zarówno w sferze produkcji jak i w systemie zarządzania jest spowodowanie przełamania podstawowych barier innowacyjnych.
Najważniejsza z nich to bariera:
— organizacyjna, wynikająca z konieczności zmiany tradycyjnych układów organizacyjnych i wiążącej się z tym konieczności zmiany podstawowych procedur działania organizacyjnego
— społeczna, wynikająca z naruszenia dotychczasowego układu ról organizacyjnych, zależności oraz systemu prestiżu i władzy organizacyjnej
— psychologiczna, wynikająca z niedostatecznego zrozumienia celu i charakteru wprowadzonych zmian oraz z charakterystycznej dla większości ludzi inercji przyzwyczajeń oraz inercji w sposobie odbierania i zrozumienia otaczającej ich rzeczywistości.
Na bazie wymienionych wyżej barier w organizacji powstaje zjawisko oporu przeciwko wszelkim zmianom.
Wśród badaczy problematyki innowacji panuje powszechnie przekonanie, że podstawową przyczyną tego oporu jest zmniejszenie poczucia bezpieczeństwa spowodowane najczęściej zagrożeniem emocjonalnym i ekonomicznym.
Interesujące ujęcie problematyki oporu przeciw zmianom daje A. S. Judson przedstawiając go w formie funkcji szeregu czynników osobowościowych.

jej realizację — bądź będą się stykać w swym działaniu z jej praktycznymi konsekwencjami.

Konflikty organizacyjne wywołane wprowadzeniem do organizacji komputera

Pojęcie konfliktu we współczesnej nauce o organizacji i zarządzaniu nie oznacza koniecznie zjawiska negatywnego.
Przykładowo Jan Zieleniewski proponuje następujące rozumienie konfliktu organizacyjnego [3].
„Konflikt wewnątrz instytucji jest to naprężenie między ludźmi lub grupami, wchodzącymi w skład zespołu, które wynika z różnic poglądów ocen albo interesów i które może, lecz nie musi wyrażać się kooperacją negatywną, może zaś — jeśli zostanie zawczasu wykryte i właściwie przezwyciężone — oraz w zasadzie powinno prowadzić do rozwoju instytucji". Rozpatrzmy jak kształtują się konflikty organizacyjne w organizacji, w której realizowany jest system informatyczny. Oprzemy się tutaj przede wszystkim na unikalnym opracowaniu tej problematyki dokonanym w oparciu o polskie doświadczenia przez W. Askanasa [4].
Zgodnie z przyjętą uprzednio zasadą podziału okresu realizacji zmiany można wyróżnić również w przypadku realizacji systemu informatycznego trzy podstawowe etapy jego wdrażania:

- prace przygotowawcze — etap ten polega na przebadaniu organizacji z punktu widzenia celów systemu, a następnie opracowanie na tej podstawie założeń systemu.
- prace wdrożeniowe — na etapie tym następuje uruchomienie poszczególnych elementów systemu oraz ich integrowanie z organizacją na zasadach określonych w projekcie
- eksploatacja systemu — jest to etap pełnego wykorzystania systemu oraz dokonywania różnego rodzaju innowacji, które spowodowane są koniecznością dostosowania systemu do zmieniających się warunków działania.

Każdy z wyróżnianych tutaj etapów charakteryzuje się innymi konfliktami organizacyjnymi.

TABELA II. Główne przejawy występowania konfliktów organizacyjnych na pierwszym etapie realizacji systemu EPD

Przejawy	Komentarze
Gloryfikowanie przez członków organizacji wykonywanej przez nich pracy oraz podawanie jej opisów w postaci jak najbardziej skomplikowanej, aby w ten sposób udowodnić niemożliwość przetransportowania jej na komputer.	Z przykładami takiego postępowania zetknęli się wszyscy autorzy badań tego problemu. Prowadzi ono w konsekwencji do niepotrzebnego komplikowania procedur komputerowych, obniżając tym samym efektywność projektowanego systemu EPD.
Fałszowanie czasokresów konieczności dostępu do informacji, tak aby wydawać się mogło, że cykl obliczeniowy powinien być bardzo często powtarzany.	Niedostrzeżenie tego typu działań prowadzi do wydłużenia czasu realizacji systemu i podrożenia kosztów jego eksploatacji.
Zbyt skąpe informowanie analityków systemu lub też przekazywanie im wręcz fałszywych informacji odnośnie do procedur i sposobów postępowania.	System po wdrożeniu produkuje zbyt wiele informacji. Działanie to, zmierzające w kierunku zachowania jak największego marginesu swobody, prowadzi w konsekwencji do konieczności wielokrotnego przerabiania systemu podnoszącej jego koszty i wydłużając czas realizacji.

TABELA III. Główne przejawy występowania konfliktów organizacyjnych na drugim etapie realizacji systemu EPD

Przejawy	Komentarze
Sprzeciw wobec spełniania wymogów EPD: czytelności, terminowości czy też dokładności informacji.	Efektem tego typu zachowań jest przedłużenie procesu wdrażania systemu oraz konieczność wielokrotnego sprawdzania informacji, co znacznie podraża całość prac.
Sprzeciw wobec konieczności uzgadniania i analizy wyników pracy systemu.	
Próby wyeliminowania grupy informatyków z przedsiębiorstwa.	Jest to przejaw walki o kompetencje, poważnie utrudniający prace nad systemem. Zjawisko bardzo typowe. W przypadku powodzenia tych prób system upada.

* * *

Obecny stan wiedzy na temat zastosowania komputerów w zarządzaniu można by scharakteryzować za pomocą jednego określenia „olśnienie możliwościami". **Staraliśmy się pokazać, że rzeczywiście możliwości komputera są ogromne i wobec wzrastającego stopnia skomplikowania procesów zarządzania w gospodarce narodowej jest konieczne jego szerokie zastosowanie w celu usprawnienia procesów informatycznych i decyzyjnych. Z drugiej jednak strony pragnęliśmy wykazać, że wprowadzenie komputera w tak skomplikowany system społeczny i techniczny jaki stanowi współczesna organizacja gospodarcza czy administracyjna wymaga szczególnej rozwagi i przygotowania nie tylko od strony technicznej.**

TABELA IV. Główne przejawy występowania konfliktów organizacyjnych w trakcie realizacji trzeciego etapu wdrożania systemu EPD

Przejawy	Komentarze
Opóźnienie w dostarczaniu informacji źródłowych i ich nierzetelność bądź nieprawidłowość.	W funkcjonującym cyklu przetwarzania jest to poważne zaburzenie funkcjonowania całości prowadzące do bałaganu organizacyjnego, a więc argument na rzecz zaniechania systemu.
Zgłoszenie dyrekcji każdego faktu nawet najdrobniejszego opóźnienia w dostarczaniu wydruków.	Jest to pochodna pierwszego typu działania, zmierzająca jednocześnie do wykazania własnego zainteresowania systemem.
Zgłaszanie zastrzeżeń do formy i treści wydruków.	Na tym etapie jest to przejaw konfliktu kompetencji połączony z próbą udowodnienia nieprzydatności systemu.
Niezgłaszanie propozycji zmian i ulepszeń.	Jest to przejaw występowania konfliktu interesu, a działanie to spowodowanie jest chęcią doprowadzenia do normalnego zużycia się systemu. Sytuacja taka wystąpiła bardzo wyraźnie w przypadkach systemów stosunkowo długo eksploatowanych.
Silne działania na rzecz wykluczania z organizacji realizatorów systemu.	Kontynuacja konfliktu kompetencji i konfliktu interesu prowadząca do zaburzeń w funkcjonowaniu systemu.

Dotychczasowe doświadczenia w tym względzie wskazują wyraźnie, że wszędzie tam, gdzie zlekceważono elementy organizacyjnego, społecznego i psychologicznego przygotowania instytucji do zmian narzuconych przez komputery, gdzie nie uwzględniono wszystkich tych czynników w fazie przygotowań, wdrażania i eksploatacji systemu informatycznego, zastosowanie komputera powodowało poważne trudności w funkcjonowaniu całej instytucji i bardzo często kończyło się niepowodzeniem.

Stąd też wydaje się, że niezbędnym warunkiem wykorzystania rzeczywistych możliwości komputera w zarządzaniu organizacjami gospodarczymi i administracją jest zrozumienie jego usługowej roli w stosunku do celów organizacji i, co za tym idzie, pozbawienia związanych z tym zagadnień swoistego nimbu tajemniczości i skomplikowania.

W tej nowej sytuacji wiele z naszych obecnych problemów ujrzymy jako dużo mniej skomplikowane i bardziej zrozumiałe w swych mechanizmach powstawania i sposobach ich likwidacji.

Andrzej Aranowski

LITERATURA

[1] G. March, A. Simon „Teoria organizacji" PWN 1969 ss. 281—282.
[2] Czesław Sikorski „Reorganizacja a usprawnienie" PWN 1975 ss. 110—140.
[3] J. Zieleniewski „Organizacja i zarządzanie" PWN 1969.
[4] W. Askanas „Elektroniczne systemy przetwarzania danych a konflikty organizacyjne wewnątrz przedsiębiorstwa". „Socjologia organizacji" Ossolineum 1975 s. 179—196.
[5] W. Askanas, — „Komputer — narzędzie czy kłopot", WEKTORY nr 17/74.

Niezupełnie a jednak w cieniu

Na cykl, jaki utworzył się już z reportaży o ośrodkach ETO, składają się wyłącznie artykuły, które powstawały na terenie miast będących już od 30 lat województwami. Ze zrozumiałych względów — przede wszystkim wskutek preferencji jakie te miasta posiadały, one najwcześniej i w najwyższej ilości były wyposażone w sprzęt komputerowy. Już jednak od pewnego czasu obok tych — nominalnie wojewódzkich istniały miasta, których predyspozycje — ot choćby z racji potencjału gospodarczego, przerastały wymiar właściwy dla większości miast powiatowych. Dziś wiele z nich już jest siedzibą województwa, ale nie dlatego po reportaż wybrałem się do Kalisza i Płocka. Raczej z zainteresowania: jak uprawiało się informatykę w dawniejszych powiatach, które były tak silne, że stać je było na komputery, a tak swoistą specyfikę, że było prawie pewnym, że wykorzystywać je musiały jakoś inaczej. A więc jak?

Dlaczego na półobrotach?

Najstarszy, jak twierdzą miejscowi i kronikarze, gród w Polsce dorobił się komputera niedawno, ośrodek przynajmniej z dekretu też jest młody. Nie znaczy to bynajmniej, że dopiero od tych formalnych dat (dostawa komputera maj 74 r., powołanie Ośrodka — czerwiec 74) poczęto się interesować w Kaliszu informatyką. Przeobrażony w 1952 r. (z reaktywowanego po wojnie Zakładu Napraw Samochodów) w Wytwórnię Sprzętu Komunikacyjnego — Zakład już w latach 65—67 oddelegował trzech hobbystów z pionu gł. technologa do nieszkodliwej „zabawy" w automatyzację obliczeń. Po krótkim kursie hobbyści zaczęli zezować w stronę Międzylesia i Wrocławia, gdzie w tym czasie stanęły pierwsze w kraju ELLIOTY 803B. Liczono tam dane określające obciążenie parku maszynowego, a później — gdy w wojewódzkim Poznaniu powstało ZETO (1967 r.) uzbrojone w MIŃSK-22, zaczęto jeździć na północ. W roku 1970 żywiołowe zapędy do informatyki poczęło koordynować Zjednoczenie Przemysłu Lotniczego i Silnikowego, zdecydowane postawić na komputery. Myślano zrazu o ROBOTRONACH, tymczasem przymiarki się przedłużały, aż dojrzała koncepcja Jednolitego Systemu. Wówczas już bez wahania zdecydowano się na RIADY. Pierwszy stanął w Rzeszowie; w lipcu 74 ruszył R-20 w Kaliszu. W okresie jego montażu i rozruchu Zjednoczenie powołało 6 Regionalnych Ośrodków EMC (Rzeszów, Andrychów, Kalisz, Świdnik, Wrocław, Mielec), z których każdy otrzymał „pod opiekę" kilka zakładów; Kaliszowi przypadły Wytwórnie Sprzętu Mechanicznego w Łodzi i Krotoszynie oraz WSK w Poznaniu.

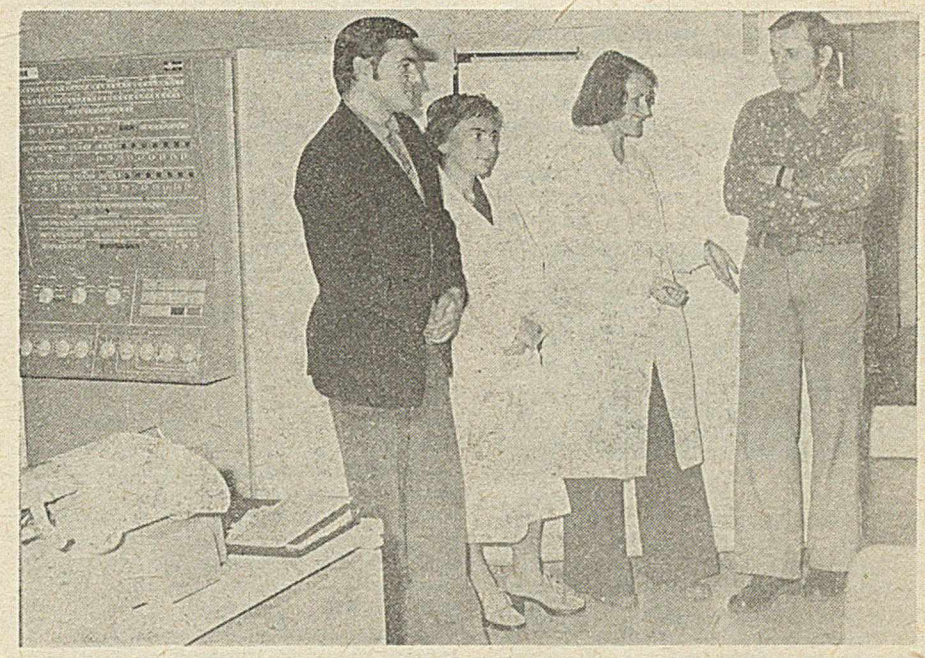


Foto 1. W koncepcji uniwersalnych systemów informatycznych dla potrzeb Zjednoczenia „PZL" Regionalny Ośrodek EMC w Kaliszu otrzymał zadanie opracowania systemu technicznego przygotowania produkcji. Pomocną młodemu zespołowi okazała się praca pracowników OBRI, który opracował STEP-74; wdrażaniem STEP-u w WSK Kalisz zajął się zespół (od lewej) mgr Bogdan Karczmasz, Krystyna Popiołek, mgr Danuta Karczmasz i mgr Czesław Szurczak (kierownik działu projektowania).

Z regulaminu Ośrodka Regionalnego (wydanego przez ZPLiS): „Podstawowym celem Ośrodka jest realizacja zadań wynikających z potrzeb informatycznych WSK Kalisz i Poznań, WSM Łódź i Krotoszyn, w ramach rozwoju Ośrodka oraz zadań wynikających z programu rozwoju informatyki w ZPLiS „PZL", (...) a przedmiotem działania Ośrodka są następujące rodzaje działalności informatycznej: a) przetwarzanie danych, b) automatyzacja prac inżynierskich i obliczeń numerycznych (...), do zakresu działania Ośrodka należy prowadzenie kompleksowych prac w zakresie: a) badań i wdrożeń systemów, b) projektowania i programowania podsystemów, c) realizacja przetwarzania danych, obliczeń numerycznych i inżynierskich, d) zabezpieczenie transmisji danych dla Centrum Obliczeniowego „PZL", e) świadczenie usług obliczeniowych dla innych instytucji w ramach wolnych mocy produkcyjnych (...), ośrodek wykonuje funkcję jednostki wiodącej w Zjednoczeniu, w zakresie podsystemu informatycznego branży — techniczne przygotowanie produkcji".

Zadanie jakie nakłada na Ośrodek powyższy kanon, byłby oczywiście niewykonalny siłami sprzed roku 1970. Ale obecnie... obecnie do dyspozycji 63-osobowej załogi (w tym: 11 projektantów, 14 programistów) stoi do dyspozycji R-20 o znakomicie rozbudo-

wanej konfiguracji: PAO — 256 kB (maksymalna dla R-20), pamięć zewnętrzna: 8 jednostek taśmowych, 8 jednostek dyskowych (po 7,25 kB), 2 czytniki kart, 2 czytniki taśmy, 2 drukarki, 2 perforatory taśmy, 1 perforator

Foto 2. Zespół decydentów OR EMC w WSK (od prawej): Marek Matuszewski — kierownik działu eksploatacji EMC, Maksymilian Gajzler — gł. analityk i mgr Andrzej Staniszczak — kierownik Ośrodka, absolwent warszawskiej SGPiS, urodzony kaliszanin powrócił do rodzinnego miasta i od roku kieruje ośrodkiem. Robi wrażenie umiarkowanego optymisty.

kart. Komputer wyposażony jest w ostatnią wersję DOS — DOS-4, obsługiwany przez 11-osobową ekipę operatorską i pod czujnym okiem 5 elektroników i 5 konserwatorów przeszkolonych na 4,5-miesięcznych kursach w Mińsku. Gdy te dane uzupełni się opisem pomieszczeń Ośrodka (mieści się w biurowcu WSK, projektowanym z uwzględnieniem potrzeb komputera

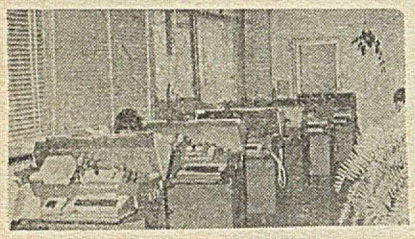


Foto 3. Sporo kłopotów przysparza brak operatorek do urządzeń przygotowywania danych, ale także i jakość tych ostatnich. Zwłaszcza do dość pracochłonnych czynności należy kontrola poprawności perforacji z uwagi na brak opisu perforacji, dziurkarki te trudno zaliczyć do nowoczesnych — stanowią element kontrastujący z klasą komputera R-20.

i ludzi) — warunki do pracy jawią się jako znakomite. Są jednak „ale"... Zaczyna się oczywiście od ludzi a sprowadza do ludzi lub sprzętu. Gdy kupowano za duże tysiące komputer, pożałowano drobnych na przetwornicę i... w wyniku spadków napięcia w okresie październik 74 — październik 75 Ośrodek „zaliczył" 154 godziny bezczynności komputera, a więc blisko 8% czasu pracy zmiany.
Gdyby więc dodać przestoje (też 8%) z powodu awarii maszyny... Brak jest pełnej obsady urządzeń do przygotowywania danych, dziurkarki zresztą też niezbyt nowoczesne (karty bez opisu perforacji)...

No i jak uprzykrzone echo powraca sprawa oprogramowania, a raczej jego niedostatku na R-20. Sporo czasu pochłonęło przeprogramowywanie tego, co ongiś pisano na MIŃSKU.
Inna słabość komputera — zbyt ogólna dokumentacja diagnostyczna. W przypadku awarii — trudno ją zlokalizować.

Ale problem największy leży gdzie indziej. Jest to układ stosunków między dyrekcją WSK a Ośrodkiem. Nie to, aby Ośrodek wadził... Tylko tyle, że odniosłem wrażenie, iż zasada partnerstwa nie jest tu realizowana. Wydaje się, że dyrekcja zbyt słabo egzekwuje z Ośrodka jego możliwości. To może wydać się dziwne, ale młoda (średnia wieku 24 lata) załoga Ośrodka nie ma ochoty siedzieć bezczynnie.

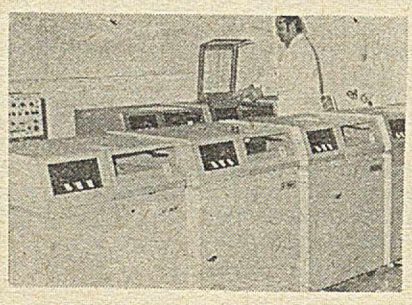


Foto 4. Kłopoty inż. Lecha Kiełtyki — głównego inżyniera, zdaje się dobiegać końca. Gdy ten materiał ukazuje się na łamach, ośrodek jest już chyba w posiadaniu przyobiecanej przetwornicy, która zlikwiduje częste przestoje wywoływane spadkiem napięcia. A kłopoty związane z awariami Riada już chyba nie mają takiego wymiaru, bowiem z upływem czasu kaliscy elektronicy coraz bardziej poznają tajniki komputera. W tle jednostki dyskowe, których ilości mógłby WSK pozazdrościć nie jeden ośrodek.

Aby więc nie gnuśnieć — pracownicy Ośrodka muszą poznawać i zgłaszać możliwości wprowadzania informatyki do systemu zarządzania WSK. Inicjatywy Ośrodka są na ogół traktowane przychylnie, ale wydaje się, że aby wreszcie nie pojawiły się symptomy frustracji — najwyższy czas, by to służby WSK zgłaszały potrzeby do Ośrodka. Zwłaszcza, że bardziej aktywny stosunek dyrekcji mógłby wyeliminować pewne dolegliwości — np. Ośrodek skarży się na brak etatów; podobno z drugiej strony gdyby nawet były — trudno by je było obsadzić, w młodych województwach, także nad rzeką Prosną informatycy nie wysiadują ławek w Urzędzie Zatrudnienia. Ale zawsze łatwiej szukać ludzi na istniejące etaty niż odwrotnie. Ten brak etatów (i ludzi) był argumentem Ośrodka na łatwo dostrzegalny fenomen — oto jedyny w Kaliszu komputer pracuje tylko na jedną zmianę. A kto by lepiej i łatwiej mógł zaspokajać potrzeby silnego przecież gospodarczo regionu (przemysł spożywczy, terenowy, handel, administracja) niż dysponujący komputerem kaliska WSK? Podobno liczne przedsiębiorstwa liczą w swoich Ośrodkach branżowych (poza terenem województwa), ale czy na pewno wypada to taniej? Brak oprogramowania na R-20 nie wydaje się być argumentem jednoznacznie tłumaczącym brak zapotrzebowania na czas komputera ze strony lokalnych przedsiębiorstw — szczupłość zespołu programistów — bardziej.
Szkoda, że dyrekcja nie dostrzega, że R-20 mógłby być kurą od złotych jajek, może wówczas znalazłyby się etaty... Bo możliwości sporego, bądź co bądź, komputera są z pewnością dotychczas nie wykorzystane, zwłaszcza że DOS-4 pozwala na pracę trójprogramową.
Podobnie jak komputer, nie w pełni wykorzystane są urządzenia do przygotowywania danych. Na 25 stanowisk (10 dziurkarek i 5 sprawdzarek kart, 10 dziurkarek taśmy) Ośrodek zatrudnia 8 operatorek.
Być może oczywiście, że za niedługi czas komputer, aby sprostać potrzebom WSK będzie obłożony pracą na pełne trzy zmiany. Indeks systemów opiewający na razie na 15 pozycji ma się niebawem powiększyć o dwa, ale pokaźny system gospodarki materiałowej przejmowany aktualnie od autora (na rzecz całego Zjednoczenia) — WSK Świdnik oraz STEP-74, system sterowania Techniczno-Ekonomicznego Produkcją wdrażany przez zespół pracowników OBRI z pionu produkcji oprogramowania, dawny R-POLSYSTEM na zamówienie WSK Kalisz, a w ogóle przewidziany jako uniwersalny dla całego Zjednoczenia.

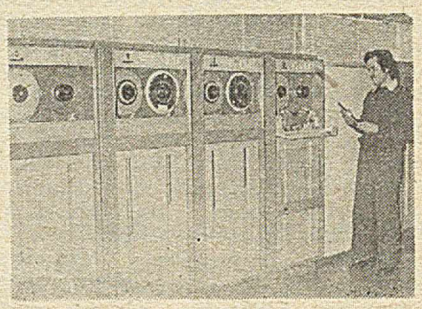


Foto 5. Mgr Ryszard Ferenc — projektant systemów, pochłonięty aktualnie systemem planowania produkcji. Jak słychać — cieszy się zasłużoną marką nawet poza sferą Zjednoczenia.

A w niedalekiej przyszłości dojdą zapewne i inne systemy — choćby płace dla już 5-tysięcznej załogi WSK (nie licząc załóg przedsiębiorstw satelitarnych), czy kartoteki magazynów (7 tys. podzespołów), ale do tego czasu komputer mógłby się zasłużyć także dla innych. Zwłaszcza, że i chęci i kwalifikacje załogi (połowa po studiach) sięgają chyba wyżej niż zamotowane dotychczas efekty. Chociaż trudno powiedzieć by nie były one widoczne. Pomimo że WSK w małym stopniu inspiruje Ośrodek, przecież bez niego już obejść by się nie potrafiła. Uruchomienie komputera rychło spowodowało trwałe zmiany w strukturze przedsiębiorstwa; jest to empirycznie sprawdzalne, zwłaszcza w sporadycznych przypadkach zapóźnienia jakiegoś obliczenia. W takim wypadku ma miejsce lawina nagłych interwencji. Należy sądzić, że z chwilą

wdrożenia STEP-u, a więc przejścia na automatyczne sterowanie produkcją — ranga ośrodka wydatnie wzrośnie. Może ruszy wówczas z miejsca sprawa etatów, może wprowadzi się nowsze techniki przygotowywania danych (choćby taśma magnetyczna) a

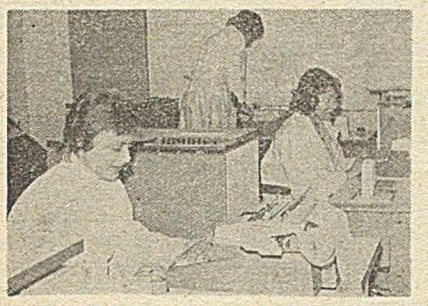


Foto 6. Urządzenia CONSUL do perforacji taśmy — uniwersalnego nośnika danych dla komputera R-20 — dopiero od niedawna w eksploatacji.

nade wszystko dostrzeże się to co przy pobieżnym oglądzie uchodzi zazwyczaj uwagi — na trudności obiektywne, możliwe do przezwyciężenia, ale kosztem dużego wysiłku pracowników, który dobrze, gdy jest doceniony. Gdy bowiem — jak mi mówiono — Ośrodek dawał wyniki z MIŃSKA, a przedsiębiorstwo widziało tylko końcowy efekt, to wkład pracy w uzyskanie takich samych wyników z RIADA będzie widziany i oceniany tylko tak samo, bez uwzględnienia czasu i wysiłku, który został zainwestowany w przeprogramowywanie systemów. Należy mieć jednak nadzieję, że kłopoty ośrodka mają się ku końcowi a jego rola oceniona zostanie właściwie.

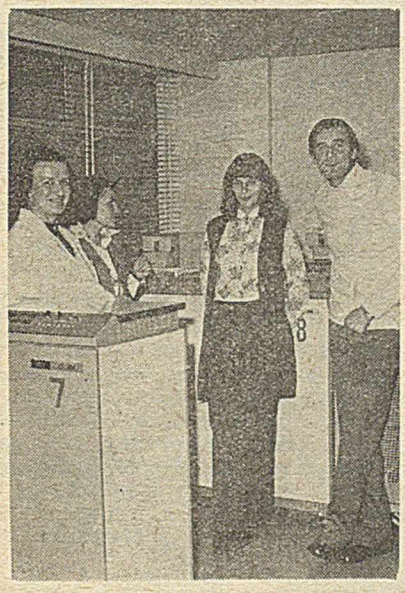


Foto 7. „Wszystko za wszystko" czyli za STEP w najbliższym czasie WSK Kalisz otrzyma system gospodarki materiałowej opracowywany na polecenie Zjednoczenia przez Regionalny Ośrodek EMC w Świdniku. Przejęcia i wdrożenia systemu w Kaliszu dokona grupa (od prawej) mgr Jerzy Gwiaździński, mgr Zofia Król, mgr Jadwiga Kościuszko, Wanda Niewiadomska.

Nieco inny niż z perspektywy Krakowa

Podobnie jak WSK Kalisz, płocka PETROCHEMIA wchodzi w skład WOG-u; płockie ZOETO powołane do obsługi macierzystego zakładu podlega szeregowi powinności dla Zjednoczenia. Niedawno, wizytując krakowski PETROINFORM, skłonny byłem mniemać, że wszystko co dzieje się w Płocku, widziane jest tak na miejscu, jak i z odległego Krakowa.

Okazuje się, że perspektywa jest nieco inna. Na miejscu w Płocku wydaje się, że specyfika Zakładu powoduje, że ZOETO modeluje styl pracy trochę inaczej niż życzyłoby sobie dążące do standaryzacji Zjednoczenie.

Podobnie jak kaliski, formalnie płocki Ośrodek jest młody, ale zbudowana 17 lat temu PETROCHEMIA już w czwartym roku istnienia przejawiać poczęła inklinacje do automatyzacji obliczeń.

Powstał dział ewidencji zmechanizowanej — późniejszy pierwszy klient ZOWARU, które to czasy dobrze pamięta ówczesny szef działu — dziś kierownik ZOETO mgr E. Rybak.

Zaczęto od gospodarki materiałowej, a gdy chcieli iść dalej, okazało się, że protoplasta dzisiejszego Zjednoczenia PETROCHEMIA nie entuzjazmuje się zbytnio ETO. Dopiero w 1972 r. powstaje ZOETO. Początkowo Zjednoczenie forsowało koncepcję centralnego Ośrodka w Krakowie silnie wyposażonego w sprzęt komputerowy z końcówkami w podległych przedsiębiorstwach. Ostatecznie zwyciężyła inna koncepcja: w jej wyniku w 1973 r. płocka PETROCHEMIA kupiła komputer HONEYWELL 2040, który w maju 1974 r. rozpoczął pracę. Po dziś dzień pracuje w tej samej konfiguracji, więc: PAO 65 k znaków (6 bitowych). Pamięć zewnętrzna 4 jednostki taśmowe, 2 jednostki dyskowe (po 39 M znaków) 1 czytnik kart, 1 czytnik taśmy, 1 drukarka wierszowa. Większym uznaniem ZOETO cieszy się o dziwo — francuski minikomputer LOGABAX 2042 wyposażony w 16 kB pamięć operacyjną, 2 jednostki dyskowej pamięci zewnętrznej (po 2,3 MB) czytnik kart, czytnik taśmy i drukarkę znakową, wsparty 10 urządzeniami ADDO do przygotowywania danych (ale ZOETO myśli o przejściu na taśmę magnetyczną). Niezależnie od centralnej koncepcji systemu zarządzania, ZOETO skoncentrowało się na:

- planowaniu produkcji (system opracowany wspólnie z Politechniką Warszawską i CHEMOAUTOMATYKĄ)
- bilansowanie produkcji
- komponowaniu paliw

Te zadania realizowane są na HONEYWELLU w pracy 2-zmianowej, zaś automatyzacja księgowości (ewidencja kosztów, finanse, rozliczanie kosztów produkcji, ewidencja środków trwałych, amortyzacja) zajmuje nieprzerwanie na 3 zmiany minikomputer LOGABAX.

Zachodnie doświadczenie torujące karierę minikomputerom, znajdują potwierdzenie w Płocku. Dominuje tu tendencja wyposażenia poszczególnych służb przedsiębiorstwa w minikomputery (zwłaszcza typu LOGABAX) ewentualnie współpracujące z HONEYWELLEM, blokująca konkurencyjną myśl rozbudowy konfiguracji komputera. Daje to oszczędności na zakupie a zwłaszcza kadrowe. Otóż ZOETO legitymując się 46 etatami (w tym 20 projektowo-programowymi) desygnuje pracowników do poszczególnych służb przedsiębiorstwa i to bezpowrotnie — jako zaczyn do tworzenia w obrębie poszczególnych służb kooperujących zespołów ETO, która z czasem w miarę narastających potrzeb będą wyposażone w minikomputery.

Prowadzić to ma w obrębie poszczególnych działów do korzystnego sprzężenia i wymiany kwalifikacji — informatyk poznaje specyfikę i zapotrzebowania problemowe działu, pracownicy — zasady korzystania ze sprzętu. Koncepcja ta znakomicie owocuje w dziale gł. automatyka, w najbliższym czasie powstanie tam autonomiczny podośrodek mający do dyspozycji 8 minikomputerów PDP-8 wsparty sterującym PDP-11, który będzie sterował automatycznie procesami nalewczymi i gospodarką wagonową (cysternową).

Totalne sterowanie procesami produkcyjnymi nie wydaje się być łatwe do wprowadzenia. W zakładzie tego typu — jak mnie informowano, — koncepcja automatycznego sterowania musi być uwzględniona już w założeniach wstępnych. Toteż ZOETO koncentruje się bardziej na procesie zarządzania, spodziewając się dojść do niego poprzez rozwiązywanie zagadnień cząstkowych i ich standaryzację na poziomie wejścia do komputera.

Z powyższych informacji wynika jednoznacznie, że niezależnie od koncepcji informatyzacji branży ZOETO w płockiej PETROCHEMII w zakresie pozostawionym do swobodnej decyzji forsuje własny model. Współpraca z ośrodkiem centralnym — PETROINFORMEM polega tu na przejmowaniu systemów opracowywanych dla branży, ale tylko tych, których opłacalność jest ewidentna. Stanie się ona jednak niebawem bardziej wszechstronna — przede wszystkim z chwilą uruchomienia łącza teletransmisji (dotychczas istnieje łączność dalekopisowa).

PETROCHEMIA jest jednym z dwóch przedsiębiorstw w Płocku dysponujących komputerami. Mimo to nie świadczy prac na zewnątrz — HONEYWELL w trybie pracy 2-zmianowej jest bez reszty wykorzystywany na potrzeby Zakładu, zaś pomimo dobrych systemów operacyjnych OS 200 i OS 2000 nie może pracować wieloprogramowo z uwagi na małą pamięć operacyjną. Tej zaś nie ma zamiaru zwiększać — rozbudowując moc obliczeniową poprzez sieci minikomputerów.

Krystyn Bernatowicz
(foto *Stanisław Kłossowski*)

Informatyka - panaceum na kłopoty transportu?

Druga już konferencja poświęcona roli informatyki w integracji systemu transportowego — INTRA-75, organizowana podobnie jak poprzednia w Kołobrzegu przez Towarzystwo Naukowe Organizacji i Kierownictwa w Szczecinie i Zakład Organizacji Przetwarzania Danych Politechniki Szczecińskiej, cieszyła się sporym zainteresowaniem kadry kierowniczej instytucji transportowych i informatyków pracujących w zagadnieniach dotyczących transportu.

Jak się zdaje staranne przygotowanie i nie błahy dorobek INTRY-74 stanowiły tak dobrą propagandę, że za koronny sympton tegorocznej, uznać należy staranną selekcję uczestników i to selekcję przeprowadzoną w jedynym możliwym po temu obszarze — w jednostkach organizacyjnych transportu. Objaw to ważny — o charakterze zasadniczym; zazwyczaj podnoszę znaczenie doboru uczestników, przeświadczony, że nawet przy najstaranniej wyselekcjonowanych referatach, cały trud staje się daremny, gdy treści te są adresowane do przypadkowych odbiorców.

Na tegorocznej, październikowej konferencji nie zabrakło przedstawicieli najpoważniejszych krajowych przewoźników (PKS — 44 osoby, PKP — 14, PSK — 3, MPK — 7) i więcej — obok przedstawicieli transportu „dalekiego zasięgu" do obrad zasiedli przedstawiciele transportu wewnętrznego (budownictwo — 23 osoby, zakłady sprzętowo-transportowe — 15 osób), zademontrowali też zainteresowanie tematem pracownicy wyższych uczelni (Politechnika Szczecińska, Akademia Ekonomiczna: Wrocław i Poznań, Uniwersytet Łódzki). Rangę konferencji podniósł fakt przyjęcia protektoratu honorowego przez wiceministra komunikacji.

Najogólniej sympozjum miało unaocznić korzyści płynące ze stosowania informatyki w procesie konstruowania jednolitego systemu transportowego — z jednej strony, zaś z drugiej dać rozeznanie czy podmiot — transport, dojrzał do wprowadzania metod informatycznych.

Sprawna organizacja INTRY-75 pozwoliła osiągnąć zamierzone cele. Jeżeli chodzi o pierwszy, tak zaprezentowane referaty jak i toczące się wokół nich dyskusje (a raczej jej brak) nie pozostawiły najmniejszych wątpliwości, co do korzyści płynących ze stosowania informatyki w sterowaniu rozmaitego rodzaju transportem; najmniej było o roli informatyki w integracji systemu, bo sama integracja jakkolwiek dziś już na pewno niezbędna jawi się trochę mgławicowo — wszyscy wiedzą, że jest potrzebna — mało kto potrafi powiedzieć jakimi drogami do niej dojść. Jeżeli zaś cel główny — integracja systemu transportowego jest dość enigmatyczny — to jak tu pokazać rolę informatyki? Stąd i postulat uczestników sympozjum o wszczęcie prac naukowych nad skonstruowaniem modelu systemu transportowego (z uwzględnieniem możliwości i roli informatyki). Że jest to osiągalne — dowodzą dobre doświadczenia niektórych przewoźników np. PKS czy TRANSBUDU. Prace teoretyków winne jednak być wsparte doświadczeniem praktyków, którzy przecież ostatecznie będą użytkownikami tak ekonomicznego jak i informatycznego systemu transportu. Aby postulowana współpraca mogła mieć miejsce i przynosić owoce jest niezbędnym aby obie strony przystały na operowanie wspólną terminologią na gruncie wspólnoty pojęć. Dotychczas da się zauważyć coś na kształt bariery, która nie pozwala na owocną współpracę.

Zaś co do możliwości stosowania informatyki w transporcie, uczestnicy INTRY, byli zgodni, że stopień możliwości zastosowania jest wprost proporcjonalny do skracania cyklu opracowywania i wdrażania systemów informatycznych.

Aby to osiągnąć należy włączyć stymulatory dla wyposażenia ośrodków (pracujących na rzecz transportu) w sprzęt i kadrę. Sprzęt — to przede wszystkim odpowiednia ilość i jakość komputerów, nowoczesne nośniki danych i sieci teletransmisji: kadra — zespoły — zwłaszcza projektantów obznajomione z zagadnieniami transportu i pracujące w ścisłym kontakcie. Mówiło się przy tej okazji o konieczności koncentracji wysiłku ludzi i mocy sprzętu na zasadniczych, kluczowych problemach.

Roli transportu dla funkcjonowania naszej gospodarki nie da się przecenić. Zresztą w najwyższych organach kierowniczych państwa, została ona w czas dostrzeżona i obecnie nie skąpi się środków i wysiłków na rzecz intensyfikacji rozwoju transportu.

W przebiegu INTRY, zwłaszcza w wypowiedziach praktyków łacno było powziąć mniemanie, że znaczne przyspieszenie w transporcie mogłoby nastąpić za sprawą nowoczesnego narzędzia decyzyjnego jakim jest komputer. Ale też i takie mniemanie, że możliwości te są jak dotychczas tylko potencjalne, że należałoby tu coś zmienić. Był to największy dorobek trzystu delegatów na konferencję TNOiK-u. (kb)

Giełda systemów INFOGRYF-76

Organizatorzy III Ogólnokrajowego Sympozjum „INFORGRYF-76" nt. „Organizacyjne problemy wdrażania systemów informatycznych" organizują giełdę systemów powielarnych jako imprezę towarzyszącą powyższemu sympozjum.

Systemy prezentowane na Giełdzie powinny spełniać następujące warunki:

— możliwość zastosowania w wielu przedsiębiorstwach

— mieć pełną dokumentację wdrożeniowo-eksploatacyjną

— być wdrożone przynajmniej w jednym przedsiębiorstwie.

Przez system powielarny organizatorzy rozumieją taki system, który poprzez zastosowanie uogólnionych rozwiązań systemowych oraz parametrycznego oprogramowania możliwy jest do wdrożenia i eksploatacji w dowolnym obiekcie ze zbiorów obiektów, dla których został zbudowany.

Ponadto systemy prezentowane na giełdzie powinny wyróżniać się następującymi walorami:

— oprogramowanie powinno stwarzać dogodne warunki do zmian organizacyjnych dokonywanych u użytkownika systemu

— zastosowanie oprogramowania użytkowego powinno być na tyle łatwe, aby odbiorca systemu nie musiał w tym celu angażować własnych, wysoko kwalifikowanych informatyków

— system powinien rozwiązyć podstawowe problemy danej dziedziny zastosowań zgodnie z obowiązującymi zasadami i przepisami ogólnopaństwowymi, resortowymi lub branżowymi

— baza normatywna systemu powinna spełniać obowiązujące wymagania w zakresie ewidencji, sprawozdawczości, analizy, metod oraz technik planowania itp.

— dokumenty źródłowe systemu muszą odpowiadać normom ogólnopaństwowym, resortowym lub branżowym.

Zasady uczestnictwa

Do udziału w Giełdzie Systemów organizatorzy zapraszają wszystkie ośrodki obliczeniowe, które opracowały i wdrożyły systemy spełniające podane wyżej warunki.

Warunkiem uczestnictwa w Giełdzie jest przygotowanie i przesłanie na adres organizatorów do dnia 30 kwietnia 1976 r., krótkiego opisu systemu (około 12 stron maszynopisu) według następującego układu:

— charakterystyka systemu (zakres, ewentualni użytkownicy, ograniczenia)

— model strukturalny systemu (podział na podsystemy i moduły)

— model funkcjonowania systemu (zbiory podstawowe, okresy przetwarzania, ideowy schemat przetwarzania)

— poziom organizacyjny systemu (sprzęt, system operacyjny itp.)

— charakterystyka wyjść systemu (z podziałem na typowe, indywidualne)

— charakterystyka systemu kodów (liczba kodów, ich struktura, ograniczenia, zakres kontroli)

— charakterystyka wejść systemu

— charakterystyka oprogramowania (indywidualne, pakietowe, parametryzowane itp.)

— określenie zasad, na jakich system może być przekazany (sprzedany) zainteresowanym przedsiębiprstwom.

Autorom opracowań zakwalifikowanych na Giełdę wypłacone będą honoraria na ogólnie obowiązujących zasadach. Informacje o systemach umieszczone zostaną w przygotowywanym „Informatorze o systemach powielarnych".

Organizatorzy przewidują nagrody dla autorów wyróżnionych opracowań (systemów).

Korespondencję należy kierować na adres:

TOWARZYSTWO NAUKOWE
ORGANIZACJI I KIEROWANIA
ul. Matejki 34
70-530 Szczecin
(tel. 447-25)

Sympozjum firmy PLESSEY

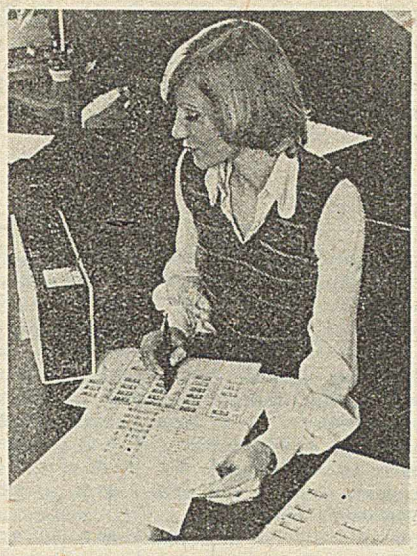
W dniach 30—31 października 1975 r. staraniem Stowarzyszenia Elektryków Polskich oraz Ośrodka Postępu Technicznego zorganizowano w Muzeum Techniki NOT w Warszawie sympozjum znanej brytyjskiej firmy PLESSEY.

Firma grupuje przeszło 250 jednostek organizacyjnych rozmieszczonych w 136 krajach Europy, Ameryki, Afryki, Bliskiego Wschodu i Australii. Zakres produkcji obejmuje szeroki wachlarz urządzeń, elementów i podzespołów elektrycznych, elektronicznych głównie dla potrzeb transmisji, rejestracji i przechowywania danych, automatyzacji procesów technologicznych i innych dziedzin. Dla zilustrowania potencjału PLESSEY warto podać, że łączne obroty w roku finansowym 1973/1974, wyniosły około 1 mld dolarów przy zatrudnieniu ponad 75 tysięcy pracowników. Z ogólnej liczby zatrudnionych na dziedzinę badań i prac rozwojowych przypada blisko 6,6 tys. pracowników. W Allen Clark, Centrum Badawczym PLESSEY, od wielu lat prowadzi się intensywne badania — między innymi nad wykorzystaniem techniki holografii w pamięciach masowych komputerów.

W czasie dwudniowego sympozjum przedstawieciele firmy wygłosili referaty na temat działalności przedsiębiorstwa oraz produkcji nowych wyrobów. Prezentowano między innymi elektroniczne centrale telefoniczne PLESSEY PENTEX, które są pierwszym urządzeniem tego typu produkowanym w Europie, centrale teleksowe sterowane komputerem (system 4660), oraz sprzęt teletransmisyjny.

Uczestnicy sympozjum mieli również okazję zapoznać się z:

- przenośnym systemem zbierania danych — PORTABLE DATA CAPTURE SYSTEM 1450



Przenośna jednostka systemu zbierania danych z piórem świetlnym i klawiaturą

- zdalną kopiarką dokumentów
- sprzętem do testowania urządzeń telegrafii (np. badania zakłóceń, pomiaru sygnałów).

Spośród demonstrowanych w czasie sympozjum urządzeń szczególnie zasługiwał na uwagę wspomniany przenośny system zbierania danych PORTABLE DATA CAPTURE SYSTEM 1450, stosowany z powodzeniem przy kontroli zapasów w przedsiębiorstwach przemysłowych i handlowych, a także w szpitalach (ewidencja chorych, kontrola wydanych lekarstw itp.) System umożliwia szybkie zbieranie danych jednocześnie z wielu źródeł (w zapisie fazowym) i zdalne przesyłanie do konwertera przetwarzającego zapis fazowy na standardowy kod akceptowany przez komputer systemu. W skład systemu zbierania danych wchodzą:

- przenośny terminal systemu zbierania danych (PORTABLE DATA CAPTURE UNIT — model 1450) wyposażony w pióro świetlne (model 1050) oraz klawiaturę numeryczną, umożliwiający rejestrację danych na taśmie magnetycznej kasetowej. Dane stałe zakodowane na specjalnych etykietach przystosowanych do odczytu automatycznego wprowadza się do systemu piórem świetlnym, natomiast klawiatura służy do wprowadzania danych zmiennych charakteryzujących przedmiot, jak np. ilość, cena, wymiary itp.
- konwerter taśmy magnetycznej (model 2920)
- drukarka etykiet (model 2101) stosowana do przygotowania wyżej wspomnianych etykiet. Drukarka ta przystosowana jest do współpracy z komputerem lub taśmą papierową
- jednostka transmisji danych (DATA TRANSMITTER — model 1670) dostosowana do standardowych modemów linii telefonicznych.

Szczególne zainteresowanie uczestników sympozjum budziło wykorzystanie systemu do kontroli wypożyczonych książek tzw. PLESSEY LIBRARY CIRCULATION CONTROL SYSTEM. System pozwala zarówno na automatyzację kontroli wypożyczanych książek w bibliotekach, jak i prowadzenie ich rezerwacji.

Do sprzętu bardzo użytecznego można zaliczyć zdalną kopiarkę PLESSEY. Kopiarka ta umożliwia zdalną transmisję dokumentów map, rysunków itp. metodą elektrostatyczną poprzez sieć telefoniczną. Pozwala ona na jednoczesne przesyłanie i odbiór dokumentów.

Z zakresu telekomunikacji firma PLESSEY oferowała również bogaty asortyment kanałów (PCM) opartych na technologii półprzewodnikowej, modemów oraz przyrządów do testowania linii teleksowych i telegraficznych.

Sympozjum wzbudziło zrozumiałe zainteresowanie. Przyczyniła się do tego, obok interesujących referatów i demonstrowanych nowych wyrobów firmy PLESSEY, sprawna organizacja, a szczególnie zapewnienie przez Stowarzyszenie Elektryków Polskich dobrej obsługi tłumaczy. (js)

ZE ŚWIATA

Ewolucja parku urządzeń do wprowadzania danych w krajach Europy zachodniej

Analiza rozwoju sytuacji w przemyśle informatyki krajów zachodnich wykazuje stały wzrost poziomu obrotów. Jednak dynamika ta charakteryzuje się tendencją malejącą. Istotny wpływ ma tu rzecz jasna panujący w świecie kapitalistycznym kryzys. Spośród dostaw różnego rodzaju sprzętu informatycznego na szczególną uwagę zasługują urządzenia końcowe. Wywierają one obecnie coraz większy wpływ na poziom aktywizacji rynku informatycznego. Stąd wielu producentów, głównie amerykańskich, przyjęło w marketingu podstawową zasadę, że „urządzenia końcowe stanowić będą siłę napędową sprzedaży komputerów". W ślad za tym idzie silna penetracja rynku światowego, a szczególnie Europy zachodniej, który zdaniem wielu specjalistów charakteryzować będzie w najbliższych latach duże zapotrzebowanie na te urządzenia, a szczególnie urządzenia do wprowadzania danych.

Według najnowszych badań grupy ekspertów Frosta i Sulivana, wartość łącznego zapotrzebowania urządzeń do wprowadzania danych dla systemów scentralizowanych wyniesie w w 16 krajach Europy zachodniej w okresie 1974—1983 ponad 6 mld dolarów. Odpowiada to przyrostowi do-

staw różnego typu urządzeń o ponad 1 mln jednostek. Wliczając do tego sprzęt, który będzie jeszcze użytkowany, stan tego parku wyniesie w 1983 roku około 1,25 mln jednostek. Warto podkreślić, że wartość dostaw tych urządzeń w podanym dziesięcioleciu jest mniej więcej równa wartości dostaw minikomputerów na wspomniane rynki krajów Europy zachodniej w tym samym okresie. Niezależnie od silnego wzrostu stanu ilościowego oczekuje się również istotnych zmian w strukturze dostaw tych urządzeń.

I tak na przykład zdaniem wspomnianych ekspertów już w roku 1976, poważne zmniejszenie dostaw dziurkartek, taśm papierowych i kart, a także jedno i wielostanowiskowych urządzeń do przygotowania danych na taśmach magnetycznych. Przy czym w tym ostatnim wypadku przełomowe będą lata 1979—1980. W roku 1976 ma także nastąpić zdecydowane obniżenie dostaw optycznych czytników pisma typu OCR *(Optical Character Reader)* czytników pisma megnetycznego MICR *(Magnetic Inc Character Recognition)* oraz OMR *(Optical Mark Reader)*.

W konsekwencji podstawową siłą napędową obrotów w tym zakresie stanie się sprzęt przygotowany do współpracy w trybie *on-line*, jak na przykład monitory (ekranowe z klawiaturą i in.), a także urządzenia do rejestracji danych na dyskach elastycznych (floppy disc) i taśmach magnetycznych kasetowych. Ewolucję struktury parku urządzeń do wprowadzania danych w systemach scentralizowanych krajów Europy zachodniej ilustruje poniższa tabela:

TABELA I

Rodzaj sprzętu	Udział procentowy jednostek		Zmiany procentowe
	1974	1980	
Urządzenia do dziurkowania kart i taśm papierowych	41,0	16,8	−24,2
Urządzenia do rejestracji na taśmach magnetycznych	23,0	19,3	−3,7
Monitowy (ekranowe i inne)	21,0	30,7	+9,7
Urządzenia do rejestracji na dyskach elastycznych i taśmach magnetycznych kasetowych	2,0	25,5	+23,5
Czytniki optyczne (OMR MICR, OCR)	13,0	7,7	−5,3
Razem	100	100	—

Jak widać największe perspektywy w systemach scentralizowanych widzi się dla urządzeń przygotowania danych na dyskach elastycznych, taśmach magnetycznych kasetowych oraz dla monitorów ekranowych. W sumie przewiduje się, że w okresie 1974—1983 łączne dostawy analizowanych grup sprzętu na rynku europejskim wyniosą:

TABELA II

Rodzaj sprzętu	Liczba jednostek	Wartość w mln dol, (ceny porównywalne
Urządzenia do dziurkowania kart papierowych i taśm	315 450	1529,0
Urządzenia do rejestracji na taśmach magnetycznych		
— jednostanowiskowe	8 550	67,0
— wielostanowiskowe	281 000	1303,0
Monitory ekranowe	285 000	1686,0
Urządzenia do rejestracji na dyskach elastycznych i taśmach magnetycznych kasetowych	155 000	978,0
Czytniki (OCR, MICR, OMR)		544,0
Razem	1 045 000	6107,0

(js)

Współpraca szkolnictwa wyższego krajów RWPG w dziedzinie informatyki

Do programu współpracy Ministerstwa Szkolnictwa Wyższego Krajów RWPG został włączony, na IX konferencji Ministrów w Hawanie, temat: „Zastosowanie techniki obliczeniowej w szkołach wyższych".

Dla określenia szczegółowej tematyki, zakresu i formy współpracy została powołana Grupa Robocza, w skład której ze strony PRL weszli: doc. dr hab. Stanisław Waligórski — MNSzWiT i doc. dr inż. Mieczysław Bazewicz — Politechnika Wrocławska.

Pierwsze posiedzenie Grupy Roboczej odbyło się w Budapeszcie w dniach 25—26.XI.1975 r.

Przewiduje się objęcie współpracą na lata 1976—1980 następujących zagadnień:

- kształcenie i dokształcanie wykładowców informatyki w szkolnictwie wyższym
- wymiana wykładowców
- wyposażenie szkół wyższych w sprzęt komputerowy jako narzędzie nauczania
- kształcenie kadr dla potrzeb informatyki. (tw)

EUROMICRO — drugie sympozjum na temat mikroarchitektury

W czerwcu 1975 roku odbyło się w Nicei pierwsze europejskie sympozjum poświęcone mikroarchitekturze systemów komputerowych. Sukces tego sympozjum spowodował, że postanowiono kontynuować tę inicjatywę organizując następne podobne sympozjum w Wenecji (na wyspie San Giorgio Meggiore) w dniach 12—14 października 1976 roku. Przy organizacji tej imprezy współpracują: IEEE Computer Society, ECl oraz European Research Office, stawiając za cel stworzenie zaawansowanego forum dyskusyjnego dotyczącego systemów mikroprogramowanych i mikroprocesorowych.

Organizatorzy proszą wszystkich zainteresowanych o nadsyłanie referatów, które powinny się koncentrować (choć niekoniecznie w sposób ścisły) w czterech następujących obszarach tematycznych (podajemy je już w brzmieniu angielskim):

- Microprogramming (techniques and applications, emulation, description simulation languages, soft-and hardware design aids, performance and reliability)
- Microprocessors and microcomputers (internal and external architectures, description and simulation languages, desing and technologies, internal interfaces and bus structures, software design tods and methodologies)
- Multimicroprocessor systems (synchronization of multimicro-processors, design, simulation and evaluation techniques, performance and reliability)
- Education (curricula, desigming with MSI and LSI, software and hardware tods).

Cztery kopie referatów (maksimum 20 stron) i streszczenie (200 słów) powinny zostać przesłane do 15 marca 1976 r. na adres: Jan Wilmink, Program Chairman Euromicro Symposium, Twente Univ. of Technology, **P.O. Box 217, Enschede 7801**, Holandia. Do 7 czerwca 1976 r. autorzy zostaną powiadomieni o przyjęciu referatów. (mh)

Informatyka na wirażu

W dniach 9—10 października 1975 r. odbyła się w Bydgoszczy kursokonferencja na temat:

„Projektowanie i wdrażanie powielarnych systemów informatycznych", zorganizowana przez TNOIK oddział w Bydgoszczy, ZETO Bydgoszcz przy współudziale Zjednoczenia Informatyki. Konferencja odbywała się pod hasłem:

„Powielarne systemy informatyczne źródłem obniżki społecznych kosztów informatyki".

W czasie dwudniowych obrad wygłoszono 8 referatów poświęconych metodom, podstawom, technikom i wdrażaniu powielarnych systemów w różnych organizacjach.

Dużą zasługą organizatorów był fakt przesłania uczestnikom konferencji materiałów w terminie, który umożliwił dokładną ich analizę. Autorzy kilku referatów, próbowali podać najistotniejsze cechy i wymogi, jakie powinno spełniać oprogramowanie, jak również szczegółowo określali etapy prac nad realizacją systemów powielarnych, do których należą:

— formułowanie zadania projektowego
— opracowanie założeń i technologii systemu
— opracowanie dokumentacji eksploatacyjnej
— konserwacja systemu.

W czasie obrad potwierdzona została teza, że etapy prac, jak również cechy systemu powielarnego są w zasadzie identyczne z cechami, jakie spełnia system indywidualny. Nie można również przyjąć za kryterium powielarności wdrożenie systemu u kilku czy kilkunastu użytkowników. Należy chyba zgodzić się z poglądem wygłoszonym przez jednego z dyskutantów, że oprogramowanie powielarne to poprostu dobre nowoczesne oprogramowanie.

Jednym z elementów, które nie sprzyjały dotychczas rozwojowi oprogramowania powielarnego był brak zasad obrotu oprogramowaniem, jak również fakt, że kolejny użytkownik zakupujący system płacił jedynie za dokumentację systemu.

Od czerwca 1975 r. obowiązuje nowy cennik usług informatycznych, który ustala cenę ponownej sprzedaży oprogramowania w wysokości 10% kosztów jego wytworzenia.

Prace nad zasadami obrotu oprogramowaniem prowadzone są nadal przez OBRI.

Jednym z poważnych ograniczeń, jakie występują w ZI we wdrażaniu systemów powielarnych to niejednolita baza sprzętowa oraz brak specjalizacji poszczególnych ośrodków obliczeniowych.

Tytuł sprawozdania „Informatyka na wirażu" wielokrotnie przewijał się w wystąpieniach dyskutantów. Wiraż próbowano zastąpić też równią pochyłą. Fakt pozostaje faktem, że wprowadzenie informatyki w wielu organizacjach nie przyniosło dotychczas spodziewanych efektów ze względu na brak dojrzałości organizacyjnej czy powolny proces wdrażania systemu.

Są jednak użytkownicy — byli tacy na konferencji, którzy uzyskali już konkretne efekty. Duże nadzieje dyskutanci wiązali także z utworzeniem Komitetu Informatyki.

Jednym z elementów dyskusji był problem nazewnictwa. Autor referatu „Podstawy metodologiczne projektowania powielarnych systemów informatycznych" proponuje zastąpienie określenia „powielarny" słowem „powtarzalny" lub „typowy". Nie wiem czy propozycja zmiany zostanie zaakceptowana przez ogół informatyków ponieważ określenie „powielarny" ma już tradycję stosowania w informatyce.

Kolejnym ciekawym elementem dyskusji był problem opłacalności wykorzystywania końcówki konwersacyjnej współpracującej „on line" z komputerem — do programowania.

Zdania były podzielone i problem nie został rozstrzygnięty czy bardziej opłacalna jest dotychczas stosowana metoda lokalnego wsadu, czy też praca z końcówki konwersacyjnej. Obydwie metody mają swej wady i zalety. Problemem tym należałoby zająć się szerzej w przyszłości, ponieważ możliwość pracy konwersacyjnej istnieje jedynie w niewielu ośrodkach, lecz wprowadzane obecnie środki techniczne Jednolitego Systemu pracę taką umożliwiają.

Na zakończenie kilka uwag i wniosków ogólnych dotyczących systemów powielarnych:

- wdrażanie systemów powielarnych jest podstawowym czynnikiem przyspieszenia rozwoju zastosowań informatyki oraz zmniejszenia nakładów na informatykę w skali kraju
- konieczne jest uregulowanie zasad odpłatności za systemy wymieniane między jednostkami różnych resortów
- generalną zasadą powinno być projektowanie jedynie systemów powielarnych, a systemy indywidualne powinny być projektowane za zgodą odpowiednich resortów.

Jerzy Kurkiewicz

Tezaurus z zakresu informatyki

Działowy Ośrodek Informacji Ośrodka Badawczo-Rozwojowego Informatyki zakończył podjętą w roku ubiegłym pracę nad opracowaniem tezaurusa [1]) z zakresu informatyki.

Tezaurus liczy ogółem ok. 1400 terminów, zawartych w trzech następujących grupach słownictwa.

Pierwszą grupę haseł podstawowych tworzą (deskryptory):
— Działy zagadnień
— Dziedziny gospodarki
— Dziedziny nauki i techniki
— Pojęcia ogólne, software
— Sprzęt (hardware)
— Tematyka.

Grupę haseł pomocniczych tworzą (modyfikatory i identyfikatory):
— Modyfikatory
— Formy informacji
— Źródła informacji
— Komputery
— Języki programowania.

Grupę haseł uzupełniających tworzą (identyfikatory):
— Język publikacji
— Chronologia
— Nazwa kraju
— Skorowidz zagadnień.

Zawartość tezaurusa w niektórych zagadnieniach wykracza poza problematykę automatycznego przetwarzania danych. Zostało to podyktowane tym, że obsługa informacyjna zakładu elektronicznej techniki obliczeniowej uwzględnia w swojej działalności szereg zagadnień z dziedziny organizacji i zarządzania, metod matematycznych, ekonomiki, metod symulacyjnych, automatyki, cybernetyki, statystyki, informacji naukowej, technicznej i ekonmicznej oraz metod i technik gromadzenia, opracowywania, przechowywania, wyszukiwania i dystrybucji informacji.

Ze względu na dynamiczny rozwój, jaki obserwuje się w dziedzinie informatyki, tezaurus będzie podlegał corocznej aktualizacji przewidując, że przyjęta struktura powinna zaspokoić potrzeby organizowanych systemów informacyjno-wyszukiwawczych w latach 1975—1980.

Wieloaspektowa struktura, którą przyjęto do budowy tezaurusa oraz dostosowana do niej symbolika — pozwolą na wprowadzanie nowych terminów, jak również usuwanie — bez powodowania zakłóceń — terminów zdezaktualizowanych.

W związku z potrzebami służb inte przewiduje się uzupełnienie kolejnych edycji następującymi wersjami językowymi: angielską, niemiecką, rosyjską i francuską.

Opracowany w Działowym Ośrodku Informacji tezaurus uwzględnia ogólnie obowiązujące postulaty i zalecenia dotyczące metodyki i organizacji budowy tezaurusów [2]).

Ujednolicona i uporządkowana terminologia stanowi punkt wyjścia dla tworzenia systemów informacyjno-wyszukiwawczych, bez których trudno sobie aktualnie wyobrazić funkcjonowanie nowoczesnej instytucji. (stz)

[1]) Tezaurus — słownik, zawierający terminy z określonej dziedziny zalecane do opisu treści dokumentu; w miarę potrzeby może on zawierać również terminy niezalecane do stosowania w tym celu z zaznaczeniem ich powiązań z terminami zalecanymi (PN-74/N-01180/00).

[2]) Zawarte m.in. w pracy: Poletyło M., Bielicka L.: Metodyka i organizacja budowy tezaurusów dla sieci inte. Warszawa 1974 IINTE Seria: Płace, Studia, Przyczynki ss. 65, zeszyt nr 1(64).

Komunikat

W iu 13.XI.197 . odbyło się spotkanie Za du Klubu gramistów z Dyrekcją Zj oczenia In matyki. Kilkugodzinna dy..usja dotyczyła warunków, jakie należy stworzyć dla podnoszenia jakości prac nad oprogramowaniem prowadzonych w ZETO, dalszego rozwoju kadry programistów i projektantów, a także niektórych problemów tego środowiska zawodowego.

Zwracano uwagę na potrzebę wypracowania kryteriów oceny produkowanego oprogramowania oraz przyspieszenia prac nad zasadami jego obrotu. Pod adresem Pionu Szkolenia OBRI wysuwano postulaty organizowania szkolenia dla wykwalifikowanych programistów i zastosowania w tym szkoleniu metod aktywnych. Wiele uwagi poświęcono współpracy OBRI z grupami programistów pracujących w ośrodkach ZETO nad opracowaniem i wdrażaniem najnowszych narzędzi w dziedzinie oprogramowania. Postulowano polepszenie wewnętrznej informacji o prowadzonych pracach.

Dyrekcja Zjednoczenia postanowiła w najbliższym czasie przeanalizować zgłoszone wnioski i postulaty i zacieśnić współpracę z Klubem Programistów. Zarząd Klubu będzie dążył do wyraźniejszego sprecyzowania roli Klubu na tle programu rozwoju ZETO oraz potrzeb środowiska projektantów i programistów. (Kw)

OŚRODKI INFORMATYKI PREZENTUJĄ

Informatyka w Fabryce Osprzętu Samochodowego w Łodzi

Zastosowania - perspektywy rozwoju

Fabryka Osprzętu Samochodowego w Łodzi jest przedsiębiorstwem średniej wielkości, specjalizującym się w produkcji zespołów, podzespołów i części dla przemysłu motoryzacyjnego.

Podstawowy asortyment stanowią grupy wyrobów:

— gaźniki samochodowe i motocyklowe
— pompy paliwowe
— sprężarki
— silniki stacyjne małej mocy
— podzespoły i elementy powietrznych układów hamulcowych
— elementy zamienne (części i podzespoły) do wszystkich wymienionych grup wyrobów.

Ogółem program produkcji przedsiębiorstwa przewiduje około 1000 pozycji, które mogą być przedmiotem zbytu.

Łączna liczba stosowanych materiałów wynosi 17 000, w tym około 4500 typorodzajów stanowią materiały podstawowe i 1200 elementy sprowadzane z kooperacji.

Liczba detalooperacji wynosi około 15 000, przy czym przeważają operacje obróbki wiórowej i montażu.

Produkcja ma charakter średnio- i wielkoseryjny o organizacji przedmiotowej, technologicznej i mieszanej.

Fabryka została objęta resortowym i branżowym programem rozwoju informatyki jak jedno z ogniw w rozwoju produkcji przemysłu motoryzacyjnego.

Dało to podstawę do rozpoczęcia prac nad obiektowym Zakładowym Systemem Informatycznym, uwzględniającym potrzeby przedsiębiorstwa oraz wymagania Resortowego i Branżowego Systemu Informatycznego.

Wyjściowym etapem prac było opracowanie „Koncepcji Rozwoju Informatyki".

ZAKRES I STRUKTURA ZAKŁADOWEGO SYSTEMU INFORMATYCZNEGO

W oparciu o przeprowadzoną analizę stanu organizacyjnego przedsiębiorstwa, szczegółowe rozpoznanie jego specyfiki i potrzeb ustalono zakres ZSI oraz jego elementy składowe.

System został podzielony na 12 podsystemów tematycznie odpowiadających podstawowym dziedzinom działalności gospodarczej przedsiębiorstwa.

Są to:

- organizacja i utrzymywanie bazy danych
- planowanie, ewidencja i kontrola realizacji produkcji
- gospodarka materiałowa
- zbyt produkcji gotowej
- zatrudnienie i płace
- koszty wytwarzania
- podsystem informowania kierownictwa
- gospodarka środkami trwałymi i przedmiotami nietrwałymi
- gospodarka narzędziowa
- gospodarka remontowa
- gospodarka finansowa
- gospodarka spedycyjno-transportowa.

W ramach każdego z podsystemów dokonano szczegółowego podziału na jednostki i elementy oraz ustalono zakres powiązań pomiędzy podsystemami. Przewidziano zintegrowanie systemu wspólną bazą danych rozbudowywaną etapowo, w miarę wdrażania kolejnych podsystemów i ich elementów.

Zakres Zakładowego Systemu Informatycznego oraz kolejność opracowań i wdrożeń poszczególnych podsystemów przedstawia rysunek 1.

Docelową eksploatację systemu oparto na sprzęcie komputerowym JS EMC.

Etapy wdrażania systemu

Etap I — realizowany jest na komputerze IBM 360/50 w Branżowym Ośrodku Obliczeniowym przy CTH Polmozbyt w Warszawie.

Korzystanie z usług ośrodka zlokalizowanego w znacznej odległości od przedsiębiorstwa ogranicza zarówno czas, jak i częstotliwość dostępu do komputera.

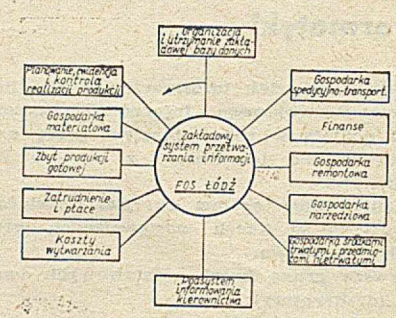
Z tych względów zakres przetwarzania został ograniczony do ważniejszych elementów systemu eksploatowanych cyklicznie w okresach miesiąc, kwartał, rok i obejmie:

— utrzymywanie podstawowych zbiorów bazy danych

— elementy podsystemu planowania, ewidencji i kontroli produkcji w cyklach rok, kwartał, miesiąc

— eksploatację na zbiorach bazy danych elementów normatywnego rachunku kosztów

— testowanie opracowywanych pakietów programów dla podsystemów: planowania i ewidencji produkcji, gospodarki materiałowej, zbytu produkcji gotowej, kosztów wytwarzania.



Baza danych

Bazę danych, której organizacja oparta jest na zaadaptowanym systemie firmowych IBM-BOMP tworzy 5 podstawowych zbiorów:

- kartoteka główna wyrobów, podzespołów, części i materiałów
- kartoteka struktury wyrobów, podzespołów i części
- kartoteka technologiczna

● kartoteka główna jednostek produkcyjnych obejmująca zbiór inwentarzowy maszyn i urządzeń, zbiór grup maszyn oraz zbiór komórek organizacyjnych przedsiębiorstwa
● kartoteka struktury produkcyjnej przedsiębiorstwa.

W oparciu o informacje zawarte w bazie danych emitowane są katalogi: indeksów wyrobów, podzespołów i części, indeksów materiałowych, technologicznych planów operacyjnych, maszyn i grup maszyn, rozwinięć strukturalnych, norm zużycia materiałów oraz szereg innych wydawnictw w układzie żądanym przez użytkowników.

Podsystem „planowanie, ewidencja i kontrola realizacji produkcji"

Uruchomiono następujące elementy podsystemu:
— ewidencja zamówień
— plany roczno-kwartalne wg asortymentu, ilości, ciężaru i wartości
— plany kwartalno-miesięczne wg asortymentu, ilości, ciężaru i wartości
— miesięczne wydziałowe plany wg asortymentu, ilości i pracochłonności
— bilans środków w zakresie potrzeb materiałowych, zatrudnienie, obciążenie maszyn dla dowolnego okresu i wariantu planu.

Równolegle z prowadzoną eksploatacją przygotowane są do wdrażania następne elementy systemu przewidziane do przetwarzania w pierwszym etapie, a w 1976 r. zostaną podjęte prace projektowe nad podsystemami gospodarki materiałowej i zbytu wyrobów gotowych.

Etap II — przewiduje dalszy rozwój Zakładowego Systemu Informatycznego w oparciu o zainstalowany w 1978 r. własny zestaw komputerowy R32. Zakłada się, że eksploatowane na IBM 360/50 zbiory kartotekowe i elementy systemu zostaną przeniesione na ten komputer.

Celem zapewnienia w pierwszym okresie eksploatacji wysokiego stopnia wykorzystania maszyny nowe opracowania będą już od 1976 r. testowane i próbnie uruchamiane na dostępnych komputerach Jednolitego Systemu. W wyniku realizacji programu prac projektowo-wdrożeniowych do 1980 r. zakresem przetwarzania zostaną objęte (poza elementami systemu uruchamianymi w etapie I) następujące podstawowe zagadnienia:

Podsystem „planowanie, kontrola i ewidencja produkcji".
— opracowywanie operatywnych miesięcznych planów produkcji dla zakładów i wydziałów
— planowanie warsztatowe — harmonogramowanie operacji technologicznych
— bieżąca ewidencja i kontrola realizacji planów obróbki części i montaży.

Podsystem „gospodarka materiałowa":
— planowanie zużycia materiałów
— normowanie zapasów materiałowych
— ewidencja obrotów i stanów materiałowych
— planowanie zaopatrzenia materiałowego
— kontrola zabezpieczenia produkcji w materiały oraz rozliczenie wydziałów z pobranych materiałów.

Podsystem „zbyt produkcji gotowej":
— ewidencja i kontrola realizacji sprzedaży wyrobów gotowych i części zamiennych
— rozliczanie magazynów wyrobów gotowych.

Podsystem „zatrudnienie i płace":
— ewidencja osobowa pracowników
— planowanie zatrudnienia i funduszu płac
— rozliczanie przepracowanego czasu i płac pracowników
— sprawozdawczość.

Podsystem „koszty wytwarzania":
— opracowanie planu kosztów
— wyliczanie i aktualizacja kosztu normatywnego
— rozliczanie rzeczywistych kosztów wytwarzania i analiza odchyleń.

Objęcie przetwarzaniem podsystemów funkcjonujących w sferze produkcji podstawowej ma dla przedsiębiorstwa podstawowe znaczenie. Kierownictwo otrzyma do dyspozycji sprawny serwis informacyjny niezbędny do efektywnego zarządzania fabryką.

Jerzy Arnold

KONCEPCJE INFORMATYKI

Co wynika z przeszłości

Ostatnie 20 lat rozwoju konstrukcji technologii komputerów charakteryzuje się czterorzędowym wzrostem szybkości liczenia, pojemności i szybkości pamięci oraz niezawodności. Na przykład w 1954 r. IBM 704 przetwarzał ok. 2 Megabitów/s, podczas gdy w 1974 r. ILLIAC IV był od niego aż 5 tys. razy a CDC STAR — 100 2,5 tys. razy szybszy.

Równocześnie gabaryty, użycie energii i koszt eksploatacji tego samego sprzętu zmniejszył się o dwa rzędy. I tak wspomniany IBM 704 przetwarzał 1954 r. za 1 dol. 1 bit/s, podczas gdy po 20 latach za tegoż samego dolara ILLIAC był w stanie przetworzyć ok. 900 bajtów/s.

Wzrost szybkości liczenia wynikał po prostu ze wzrostu szybkości podstawowych obwodów, zmniejszenia gabarytów samego sprzętu oraz z innowacji w architekturze sprzętu. Dodatkowo, innowacje w oprogramowaniu, jak na przykład wieloprogramowość i podział czasu, spowodowały lepsze wykorzystanie możliwości obliczeniowych, co w konsekwencji zwiększyło przelotowość *(throughput)* zadań przetwarzaniowo-obliczeniowych. Natomiast zmniejszenie gabarytów, zużycia energii, ciężaru i kosztu wynikało z miniaturyzacji obwodów, masowej produkcji i jej automatyzacji oraz zautomatyzowanego projektowania.

Ulepszenia i wzrost wydajności komputerów pojawiały się w modnym, cyklicznym trybie, gdzie cykl określano generacją.

„Cykl życia" każdej generacji przechodził przez kolejne etapy. Przekazywana z laboratorium do produkcji nowa technologia układów była na ogół o rząd wielkości bardziej efektywna od poprzedniej. Równocześnie były ulepszane metody projektowania i wytwarzania, co pociągało za sobą wzrost wydajności, efektywności, zmniejszanie kosztu itp. Równocześnie w laboratoriach powstały nowe technologie układów, nowe koncepcje architektury komputerów, które mogły być realizowane dopiero w następnej generacji.

Definicja „generacji komputerowej" nie jest precyzyjnie sformułowana. Istnieje wiele poglądów co do liczby generacji, które wystąpiły i co do kryteriów ich wyróżnienia.

Termin „generacja" był szczególnie modny w latach 1965—1970, kiedy komputery IBM 360 miały być tymi jedynymi właśnie z III generacji. Potem szereg firm zastosowało w swoich maszynach obwody scalone i zaczęło się chwalić produkowaniem komputerów III generacji. Miało to oznaczać przepustkę do nowoczesności. Tymczasem architektura tych maszyn właściwie nic nowego nie wnosiła by mówić o jakimś zasadniczym przełomie.

Z tego względu powinniśmy raczej mówić o generacjach obwodów i generacjach komputerów. Albo też zgodzić się, że o generacji sprzętu informatycznego decyduje wektor dwuskładnikowy (obwody i komputery).

Zgodzono się z następującą charakterystyką generacji obwodów:

Generacja O. Przekaźnikowa. Zbudowano takie maszyny jak HARDVARD — IBM, ENIAC (z lampami próżniowymi). Okres do 1953 roku.

Generacja 1. Lampowa. Z rynkowych maszyn można wyróżnić: UNIVAC I, IBM 701, IBM 704, IBM 709. Okres 1951—1958.

Generacja 2. Tranzystorowa, obwody drukowane. Zaliczymy do nich takie maszyny jak: PHILCO 2000 IBM 7090 (niegdyś najpopularniejszy komputer uczelniano-laboratoryjny), CDC 6600 (następca poprzedniej maszyny w uczelniach) oraz wielkie maszyny, jak

STRETCH (IBM 7030) czy UNIVAC LARC. Do generacji tej należą nasze ODRY 1304 i ZAM 41, radzieckie MIŃSKI 22 i 32 i inne. Okres 1958—1969.

Generacja 3. Mikroscalona. Przykładem jest tu IBM 360, BURROUGHS 6500, UNIVAC 1108, ODRA 1305, Jednolity System EMC. Okres od 1965 do chwili obecnej.

Generacja 4. Średnioscalona. Zastosowane zostały obwody MSI, LSI. Przykładem może być tu ILLIAC IV lub doświadczalny AADC *(All Application Digital Computer,* skonstruowany dla potrzeb marynarki amerykańskiej).

Następna generacja nie została jeszcze zdefiniowana. Sugeruje się dla niej nazwę, „Wielkoscalona", jako że w grę mają wchodzić obok obwodów LSI również GSI *(Grand Scale Integration).* Obwody mają być monolityczne już na poziomie molekularnym. Generacja 6 ma wykorzystywać hybrydowe układy monolityczne *(solid state)* — holograficzne, jakkolwiek szereg prognoz widzi w nadchodzącej dekadzie lat osiemdziesiątych dalsze wykorzystywanie układów średnioscalonych (generacja 4).

Podział maszyn na generacje komputerowe służy nie tylko celowi rozróżnienia tych własności komputerów, jakie wynikają z zastosowania określonych obwodów, ale również tych, jakie wynikają z określonej architektury i oprogramowania. Podział ten może służyć rozróżnieniu maszyn, które należąc do jednej generacji układów, różnią się na przykład 10-krotnie w wydajności. Mam tu na uwadze PHILCO 212 i CDC 7600. Aczkolwiek wysoka wydajność CDC 7600 wynika z zastosowania szybszych tranzystorów, to jednak architektoniczne innowacje sprawiają, że ta tranzystorowa maszyna jest wydajniejsza od IBM 360/91 na obwodach scalonych. Właśnie owa maszyna CDC zwróciła uwagę na w gruncie rzeczy drugorzędne znaczenie techniki układowej w stosunku do rozwiązań organizacyjno-programowych. Podobnie ma się sprawa z tymi samymi komputerami, które choć wykonane w różnych technikach niekoniecznie reprezentują różne generacje komputerów.

Dobrym tego przykładem jest ODRA 1304 i ODRA 1305.

Generacje komputerowe można scharakteryzować następująco. Generacja 1. Komputery specjalizowane, wprowadzone w latach 1951—1952 dla wykonywania pojedynczych naukowych i ekonomicznych zadań obliczeniowych, z około 100 prostymi rozkazami, sporadycznymi rejestrami indeksowymi, podprogramami oraz prostym językiem symbolicznym.

Generacja 2. Komputery uniwersalne, wprowadzone w latach 1958—1960 dla ogólnego przetwarzania danych, z ok. 100 złożonymi rozkazami, niezależnym i współbieżnym wykonywaniem operacji wejściowo-wyjściowych, szybką wewnętrzną i masową pamięcią zewnętrzną, językiem programowania wyższego rzędu oraz rozbudowanym oprogramowaniem podstawowym przejmującym większość czynności operatorskich i ukierunkowanym na przetwarzanie partiowe.

Generacja 3. Systemy albo rodziny komputerów, wprowadzone w 1963—1965 dla ogólnego przetwarzania danych, wieloprogramowe, z podziałem czasu w wykonywaniu operacji w jednostce centralnej, ze zdalnymi urządzeniami końcowymi do przetwarzania konwersacyjnego, a także systemy wieloprocesorowe ukierunkowane głównie na przetwarzanie bieżące z rozbudowanymi systemami operacyjnymi, wyposażonymi w wiele języków programowania oraz wewnętrzną bibliotekę programów modularnych.

Generacja 4. Sieci systemów komputerowych, wprowadzone w 1970—1972 dla bieżącego i bezpośredniego *(on-line)* przetwarzania informacji, wieloprocesorowe, złożone z mini- i mikrokomputerów, z podprogramami zrealizowanymi sprzętowo, mikroprogramowalne, wyposażone w metakompajlery, otwarte języki programowania itp.

Podana lista własności trzeciej i czwartej generacji komputerów nie wyczerpuje wszystkich. Oczywistym jest trend w kierunku rozwoju oprogramowania upraszczającego użytkowanie sprzętu. Rozkwit oprogramowania, jaki obserwujemy od 3 generacji komputerowej zawiódł oczekiwania. Chyba dlatego, że sam sprzęt tej generacji wymaga takiej a nie innej komplikacji oprogramowania. Koncentrycznie zorientowane zestawy komputerowe prowokują do kierowania danych i obliczeń do głównych i najwyżej postawionych w hierachii systemu jednostek centralnych. Wystarczy przyjrzeć się oprogramowaniu IBM-owskiemu bazy danych (IMS).

Ani projektant ani programista nie ma wyboru, wszystko idzie do „centrum". Oprogramowanie musi skoordynować działanie dziesiątek a nawet setek końcówek, które połączono z owym „centrum". Połączono całkiem niepotrzebnie. Rośnie koszt trnsmisji danych, oprogramowania, sprzętu, specjalistów itd. Systemy takie dysponujące coraz większą mocą obliczeniową, a coraz słabiej wykorzystywane zadaniami użytkowymi — przekształcają się w zbiurokratyzowane giganty przetwarzające programy wewnętrznej koordynacji. Dodatkowo nakłada się potrzeba rozbudowanej kontroli dostępu do zbiorów i programów. I dlatego uproszczenie użytkowania sprzętu można osiągnąć dopiero w 4 generacji komputerowej. Oczywiście sprawa ta jest niezmiernie kłopotliwa dla dotychczasowych producentów sprzętu „tradycyjnego".

Konserwatywnie nastawieni, niezdolni jeszcze do opanowania oprogramowania sprzętu, który dotychczas produkują — muszą zająć stanowisko wobec nowej sytuacji. W warunkach kiedy użytkownik może dokonać swobodnego wyboru sprzętu, dochodzi do rozwiązań drastycznych w postaci bankructwa firm. Rok 1975 przyniósł dwa, a mianowicie francuskiej CII i amerykańskiej filii oddziału komputerowego XEROXA. Mówi się o nieuniknionych kłopoach firm CDC i NCR, które może spotkać los firm uprzednio wymienionych. Od dwóch lat HONEYWELL zwalnia personel, bowiem nie może dać sobie rady z przypadkową mieszanką programu produkcyjnego powstałą w wyniku fuzji w 1972 roku z oddziałem komputerowym firmy GE. W firmie SINGER zaszła najbardziej znamienna zmiana, kiedy prezesem jej oddziału komputerowego został w październiku 1975 r. G. Cogar, dotychczasowy szef firmy produkującej minikomputery COGAR 1500. Okazało się, że produkowany przez SINGERA komputer średniej klasy o nazwie SYSTEM 10 nie daje tych szans koncernowi co wspomniany minikomputer, na bazie które powstało już kilka rozwiązań sieci komputerowych we Francji i w Belgii.

Nawiasem mówiąc casus dr Cogara jest znamienny dla obecnego stanu informatyki po stronie producentów. Otóż jest on chyba obecnie jedynym fachowcem na tak wysokim stanowisku w amerykańskim przemyśle komputerowym. Współtwórca UNIVACA I, III (które przed IBM dały początek zastosowaniom gospodarczym komputerów) oraz maszyn PHILCO, założyciel firmy DATA GENERAL (minikomputery NOVA), założyciel firmy MDS (pioniera w budowie klawiaturowych rejestratorów danych na nośnikach magnetycznych), pierwszy producent pamięci na obwodach scalonych, a wreszcie konstruktor i założyciel firmy COGAR, produkującej minikomputer 1500 przystosowany do komponowania sieci komputerowych.

Okazało się, że wiele firm komputerowych było bardzo źle zarządzanych, przez tzw. „ogólnych biznesmenów", których wiedza „wyjściowo-wejściowa" umożliwiała im prowadzenie firm tylko wtedy, gdy miały one dobry rozpęd początkowy. Co wybitniejsi fachowcy przenieśli się na uczelnie (np. F. Brooks, twórca IBM 360) bądź założyli własne firmy konsultacyjne (np. współtwórca szybkich maszyn CDC).

Dziesiątki innych założyło własne małe firmy projektowo-produkcyjne, których liczba sięga dziś 5 tysięcy. Firmy te skutecznie zachwiały dotychczasowymi kolosami, obnażając przede wszystkim ich brak fachowego rozeznania. To samo dzieje się w IBM. Fakty wykazały, że najwięcej sprzedano nie komputerów 360/370 lecz małych komputerów SYSTEM 3, który jest obecnie najpowszechniej spotykanym komputerem IBM, a co za tym idzie również najpopularniejszym komputerem w skali światowej.

Wypuszczona w 1975 r. dalsza jego odmiana określona jako FUTURE SYSTEM, a mianowicie System 32 ma wg przewidywań firmy zapewniony zbyt rzędu setek tysięcy egzemplarzy. Prognoza wewnętrzna firmy IBM do 2050 roku (sic!) przewiduje zbyt około 100 mln komputerów. Czytelnik zdaje sobie chyba sprawę z tego jakie będą owe komputery. Na marginesie tych rozważań zwróćmy uwagę, że System 32 jest omal kopią wspomnianego mini systemu SINGER-COGAR 1500.

Potwierdza to zresztą tylko tezę, że firma IBM nigdy nie wychodzi sama z nowinkowym sprzętem, a tylko swymi rozwiązaniami uogólnia i potwierdza dotychczasowe rozwiązania, które już sprawdziły się na rynku.

Dla potwierdzenia tej tezy sięgnijmy po następny przykład z pamięcią wirtualną, którą tak mocno spopularyzowały maszyny serii IBM 370. Otóż rozwiązanie to było znane w paru innych komputerach takich choćby jak LARC, ICT 1906 czy GAMMA 60 (sprzed lat 16).

By zakończyć nasze rozważania o generacjach, warto zauważyć, że jak dotąd pojawiły się one co 6 lat. Prawie każda nowa generacja obwodów powodowała następujące zmiany w charakterystykach maszyn: prędkość liczenia wzrastała 10 a pojemność pamięci 20 razy, niezawodność podniosła się 10-krotnie, wreszcie koszt układów malał 10 razy, a koszt systemu obliczeniowego 2,5 raza. Może w okresie szukania efektywności naszych systemów informatycznych tego typu analizy odnośnie do sprzętu krajowego można by uznać za najbardziej potrzebne.

MAT

TERMINOLOGIA

Poniżej przedstawiona została propozycja terminologii w zakresie struktur danych. Autorzy zdając sobie sprawę z jej niedoskonałości będą wdzięczni za uwagi i ewentualne inne propozycje. Przedstawiona terminologia została zaczerpnięta (z minimalnymi zmianami) z pracy pt.: „Wybrane zagadnienia związane ze strukturami danych" opublikowanej w Pracach IMM, zeszyt 1/1974, str. 7—75.

ADMINISTRATOR BAZY DANYCH
(data base administrator)
Osoba lub grupa osób zapewniających ciągłą eksploatację bazy danych

BANK DANYCH
(data bank)
Baza danych wraz z oprogramowaniem; oprogramowanie powinno w szczególności zapewnić niezależność programów od struktury zapisu oraz możliwość jednoczesnego korzystania z bazy danych przez wielu użytkowników

BAZA DANYCH
(data base)
Zbiór zapisany w pamięci maszyny w ten sposób, że każda dana występuje raz; w bazie danych muszą być wyróżnione podzbiory (niekoniecznie rozłączne) danych równoważnych z punktu widzenia ich tajności

BLOK
(block)
Jednostka wymiany między pamięcią główną a pomocniczą

DANA
(datum)
Para (nazwa, wartość) gdzie wartością może być stała, dana lub struktura

DANA PROSTA
Dana, której wartością jest stała

DOKUMENT
(logical record, entry)
Dana, której wartością jest zbiór pozycji i/lub grup

DOSTĘPNOŚĆ
(privacy)
Zespół środków umożliwiających użytkownikowi dostęp do wybranych danych, procedur itp. (zob. też ZABEZPIECZENIE)

ELEMENT ZAPISU
(data element, field)
Jednostka pamięci głównej pobierana lub pamiętana w czasie jednego cyklu. Może być bitem, bajtem, słowem

GŁÓWNA METODA DOSTĘPU
Metoda dostępu wyznaczająca organizację zbioru danych

GRUPA, POZYCJA ZŁOŻONA
(group, data aggregate)
Dana, której wartością jest zbiór pozycji i/lub grup

IDENTYFIKATOR
(identifier)
Klucz dokumentu jednoznaczny w pliku

KLUCZ DOKUMENTU
(record key)
Jedna z pozycji w dokumencie

KLUCZ PORZĄDKOWANIA
(major key)
Klucz dokumentu wyznaczający miejsce dokumentu w pliku

KLUCZ WYSZUKIWANIA
(minor key)
Klucz dokumentu służący do odszukania dokumentu w pliku

METODA DOSTĘPU
(access method)
Algorytm znajdowania zapisu lub miejsca na zapis. Do najbardziej znanych metod dostępu należą metody dostępu: sekwencyjnego (serial lub sequential), indeksowo-sekwencyjnego (index-sequential), bezpośredniego (random)

OCHRONA
(protection)
Zespół środków zabezpieczających przed przypadkowym lub umyślnym zniszczeniem zbioru danych, procedur itp. oraz przed niepowołanym do nich dostępem (zob. też ZABEZPIECZENIE)

OPEROWANIE DANYMI
Wykonywanie operacji na danych. Operacje na danych można podzielić na dwie klasy: operacje użytkownika oraz operacje na zbiorach danych

ORGANIZACJA PLIKU
(file organization)
Powiązania w pliku (np. między dokumentami lub grupami) deklarowane przez użytkownika

ORGANIZACJA ZBIORU DANYCH
(file organization, data set organization)
Powiązania między obrazami dokumentów w pamięci wyznaczone przez główną metodę dostępu. Najbardziej znane organizacje: sekwencyjna (serial lub sequential), indeksowo-sekwencyjna (index-sequential), bezpośrednia (random lub direct) inaczej zwana organizacją o dostępie obliczanym (computed access), dowiązywana (chained) inaczej zwana sieciową (network), odwrócone: częściowo i całkowicie (partial i full inverted) oraz grupowa (clustered)

PLIK
(file)
Dana, której wartością jest zbiór dokumentów

PLIK FORMATOWY
(formatted file)
Plik, w którym dokumenty mają takie same nazwy pozycji; pozycje występują w tej samej kolejności, a odpowiadające sobie pozycje mają wartości o takiej samej długości

PLIK NIEFORMATOWY *(unformatted file)*	Plik, w którym dokumenty mogą mieć różną liczbę pozycji
PLIK PÓŁFORMATOWY *(semiformatted file)*	Plik różnicy się od pliku formatowego tym, że wartości pozycji mają nieokreśloną (zmienną) długość
POZYCJA, POZYCJA ELEMENTARNA *(data item, data element)*	Dana prosta, podstawowa jednostka w przetwarzaniu danych
PRZENASZALNOŚĆ *(portability)*	Możliwość przenoszenia oprogramowania (danych) między dwoma systemami
RELACJA SORTUJĄCA	Relacja spójna, zwrotna i przechodnia
STRUKTURA DANYCH *(data structure)*	Zbiór danych i/lub stałych
STRUKTURA ZAPISU *(storage structure)*	Obraz struktury danych w pamięci maszyny
TOM *(volume)*	Zbiór bloków dostępny z urządzenia we—wy w pojedynczym ładowaniu; może zawierać część zbioru danych bądź jeden lub kilka zbiorów danych
WARTOŚĆ PROSTA, STAŁA *(simple value, constant)*	Wartość niepodzielna w języku (nie zawierająca żadnych innych danych)
ZABEZPIECZENIE *(security)*	Zespół środków ograniczających operowanie danymi, procedurami itp. do określanego kręgu użytkowników. Zabezpieczenie składa się z ochrony i dostępności (zob. też DOSTĘPNOŚĆ i OCHRONA)
ZAPIS *(physical record)*	Zbiór powiązanych ze sobą elementów zapisów, który można zidentyfikować, najczęściej obraz dokumentu
ZBIÓR DANYCH *(data set)*	Zbiór bloków mających wspólną nazwę, najczęściej obraz pliku.

Danuta Kołacka, Anna Wieczorek, Jan Wierzbowski, Julian Winiewski

PROBLEMATYKA BAZY DANYCH

Uwagi do koncepcji Uniwersalnego Systemu Zarządzania Bazą Danych dla komputerów JS EMC

Dzieło wystawione na widok publiczny jest prowokacją do krytyki. Skończone dzieło może być przedmiotem pozytywnej lub negatywnej oceny, natomiast celem publikacji koncepcji dzieła jest zwykle nie tylko uzyskanie aprobaty środowiska, ale też zebranie konstruktywnych propozycji i wniosków, które mogą ulepszyć ostateczny kształt dzieła.

Taki cel — wyraźnie sprecyzowany — przyświecał również autorom Koncepcji Uniwersalnego Systemu Zarządzania Bazą Danych (USZBD) dla komputerów JS, opublikowanej na łamach Informatyki [1], [2]. Wprawdzie Autorzy zapowiedzieli serię dalszych artykułów na ten temat, ale opis języków SZBD zamieszczony w numerze 7—8 Informatyki już pozwala — mimo fatalnego złamania artykułu i licznych błędów korektorskich, o czym dalej — wyrobić sobie pogląd na całościową koncepcję proponowanego systemu.

Jako konstruktor bazy danych, posługujący się w tym celu narzędziem: systemem zarządzania bazą danych, pragnę ustosunkować się do propowanego nowego narzędzia, którym być może będę posługiwał się w przyszłości.

Może na początek kilka truizmów. Wartość użytkowa, a zresztą i wartość pieniężna systemu cyfrowego zależy nie tylko od konstrukcji, logiki i konfiguracji sprzętu *(hardware)*, ale również i w coraz większym stopniu od poziomu oprogramowania *(software)*, które umożliwia różne zastosowania komputerów.

Wiadomo, że centralnym punktem, powiedziałbym — sercem nowoczesnego systemu informatycznego jest Bank Danych (niekoniecznie scentralizowany wg. niektórych nowych, ale kontrowersyjnych koncepcji). Bank Danych składa się z dwóch elementów: bazy danych i systemu zarządzania bazą danych. SZBD jest narzędziem tworzenia, utrzymywania i wykorzystania bazy danych, stosownie do życzeń użytkowników sformułowanych w programach zastosowań lub wyrażeniach języka konwersacyjnego. Nie można zatem przystąpić do konstrukcji bazy danych, jeżeli nie dysponuje się w tym celu odpowiednim narzędziem.

To stwierdzenie określa wagę i znaczenie prac podjętych przez autorów koncepcji uniwersalnego SZBD dla maszyn Jednolitego Systemu. Na nic się nie zda nawet najlepsza myśl konstrukcyjna pracowników ELWRO i innych producentów maszyn rodziny RIAD, jeżeli nie będzie stosownego oprogramowania.

Autorzy słusznie oparli się na propozycjach DBTG CODASYL, których znaczenie i coraz szersze implementacje sami opisali w artykule [3]. Rodzi się tu jednak pierwsza wątpliwość. Zwykle implementacje są nie tylko dostosowaniem do wymagań i ograniczeń konkretnego systemu cyfrowego lub zastosowania, ale też są pewnym zawężeniem funkcji wzorca. Po prostu w implementacjach np. języka COBOL, ALGOL i wszelkich innych, nie korzysta się z całego asortymentu możliwości zawartych we wzorcu. Daje to w rezultacie może mniejszą uniwersalność i elastyczność, ale zwiększa łatwość posługiwania się się wersją implementowaną.

Autorzy koncepcji USZBD podeszli do Raportu CODASYL w sposób maksymalistyczny: nie tylko w pełni wykorzystują propozycje DBTG, ale jeszcze starają się rozszerzyć funkcje systemu. W rezultacie powstałby rzeczywiście bardzo uniwersalny system, niezwykle elastyczny, dający projektantowi bazy danych możliwość odwzorowania wszelkich logicznych

związków między obiektami realniej, otaczającej nas rzeczywistości. Taki system nadawałby się do bardzo wielkiej klasy zastosowań. Chodzi jednak o to, czy stać nas na taki system i czy jest on rzeczywiście potrzebny w tej skali możliwości, przynajmiej w pierwszej wersji.

Po pierwsze — uniwersalność kosztuje. W tym wypadku jest to czas i koszty opracowania i wdrożenia systemu, bardzo wielki obszar w pamięci operacyjnej zajmowany przez SZBD, koszty bieżącej eksploatacji itp.

Nie wiem, czy w naszych polskich warunkach, przy tak szczupłej kadrze informatyków mających jakiekolwiek praktyczne doświadczenie w projektowaniu i oprogramowaniu tej klasy zadań, istnieje realna szansa zakończenia sukcesem tego rodzaju przedsięwzięcia. Pakiet programów składający się na SZBD winien być harmonijnie powiązany nie tylko z systemem operacyjnym i całym oprogramowaniem, ale również wykorzystywać logiczne możliwości sprzętu.

Po drugie — osobiście posługuję się implementacją propozycji CODASYL na maszynę UNIVAC 1100. Jest ona znacznie ograniczona w stosunku do wzorca, np. istnieje tylko możliwość łańcuchowego wiązania rekordów w *setach* [1]), a nie stosuje się powiązania za pomocą tablic odsyłaczy (POINTER ARRAY). Mimo to zawarte w tej implementacji możliwości deklarowania różnych strategii lokowania i wyszukiwania rekordów w bazie danych sprawiają trudność wyboru strategii dla konkretnego zastosowania w praktycznym konstruowaniu bazy danych. Proponowany w USZBD asortyment strategii może dla początkujacych projektantów bazy danych zakończyć się znanym powiedzeniem: „osiołkowi w żłobie dano..." (przepraszam, nie chcę tu nikogo obrażać), lub może prowadzić do przypadkowego wyboru strategii, niepotrzebnie zwiększającej czas przeprzetwarzania lub ogromnie zwiększającej niewykorzystane przestrzenie w pamięci zewnętrznej, albo też powodującej zjawisko „migotania stron".

Cała wartość banku danych zależy od takiej konstrukcji bazy danych, która pozwoli w możliwie najprostszy sposób i w najkrótszym czasie pozyskać zapisane w niej dane, co zależy właśnie od strategii lokowania i wyszukiwania danych.

Po trzecie — Anglicy mówią, że dobry Administrator Bazy Danych jest na wagę złota. Jest to pierwszy i podstawowy użytkownik systemu zarządzania bazą danych. Zadania jego i funkcje są tak skomplikowane, że nie wiem, czy należy jeszcze bardziej je komplikować dając do ręki programistom użytkowym tak uniwersalne narzędzie, za pomocą którego będą oni mogli wpływać na konstrukcje bazy danych. I nic tu nie pomoże nieufność w stosunku do programistów i wyposażenie Administratora Bazy Danych w narzędzie kontroli ich działalności przy pomocy klauzuli ON.

Po czwarte — w koncepcji USZBD bardzo rozbudowano system procedur bazy danych i funkcji pomocniczych, których wypełnienie konkretną treścią pozostawia się projektantom konstruującym bazę danych i opisujących jej strukturę. W implementacji UNIVAC DMS-1100 ewentualnie wymaga się opracowania tylko jednej procedury bazy danych, mianowicie procedury CALC, i okazuje się, że wcale nie jest tak łatwo opracować tę procedurę optymalną dla danego zastosowania.

Wydaje się więc, że proponowany uniwersalny SZBD jest zbyt skomplikowany dla posługiwania się nim przy konstruowaniu bazy danych i może nasuwać wiele trudności szczególnie początkującym projektantom, a przecież nie chodzi o narzędzie dla wybranej elity projektantów baz danych. Czy nie należałoby zatem nieco ograniczyć ambicje i pójść raczej na opracowanie Systemu Zarządzania Bazą Danych mniej uniwersalnego, ale prostszego do wdrożenia i łatwiejszego w posługiwaniu się nim?

Drugą wątpliwość nasuwa filozofia przeniesienia logicznego zapisu bazy danych na fizyczną reprezentację (strukturę zapisu) w pamięci zewnętrznej.

W tym celu Autorzy opracowali specjalny Język Opisu Pamięci, który poprzez zdanie DD z *Job Control Language* wiąże opis struktury logicznej bazy danych z systemem operacyjnym, który ustala konkretne urządzenie pamięci zewnętrznej i bezwzględny adres reprezentacji fizycznej obszaru bazy danych.

Proponowany Język Opisu Pamięci jest zbliżony do implementacji UNIVAC, z tym, że tam stanowi on część składową Języka Opisu Danych *(Data Definition Language)*. Projektant bazy danych posługuje się więc tylko jednym językiem, który zapewnia opis nie tylko logicznej struktury danych, ale jednocześnie przeniesienie jej na fizyczną reprezentację w bazie danych. To przeniesienie odbywa się za pomocą utrzymywanego przez system operacyjny MASTER FILE DIRECTORY, służącego do rejestracji stałych zbiorów systemu. Obszary Bazy Danych są traktowane jak wszystkie inne zbiory sterowane systemem EXECUTIVE i w MASTER FILE są zapisane ich charakterystyki i aktualne adresy reprezentacji fizycznej, działa tu bowiem automatyczny mechanizm ROLLIN/ROLLOUT.

Dla projektanta bazy danych jest to bardzo przejrzysta i prosta forma opisu logicznej i fizycznej struktury bazy danych. Zastosowanie odrębnego Języka Opisu Pamieci wymaga dodatkowych przebiegów translacji i kompilacji, dodatkowego testowania opisu bazy danych, mechanizmu łaczenia opisu fizycznego z opisem logicznym, dodatkowego obszaru w pamięci operacyjnej przy przetwarzaniu bazy danych itp. itd. Może więc pójść na mechanizm podobny do zastosowanego w implementacji UNIVAC DMS-1100, uprościć całe zagadnienie i utworzyć jeden prostszy Język Opisu Danych? Rozumiem, że taka implementacja Raportu CODASYL jest znacznie trudniejsza, ale przecież w ostatecznym rachunku chodzi o ułatwienie życia projektantom bazy danych, a nie projektantom systemu zarządzania bazą danych.

Były to najbardziej ogólne uwagi dotyczące samej filozofii proponowanego systemu, a teraz garść uwag w sprawie konkretnych rozwiązań.

1. W implementacji UNIVAC DMS-1100 klucz bazy danych zajmuje stałą pojemność jednego słowa (36 bitów) i składa się z trzech elementów: numer obszaru w bazie danych, numer strony w obszarze i numer rekordu na stronie. Klucz bazy danych jest więc bezwzględnym adresem logicznym rekordu w bazie danych i w tym charakterze jako odsyłacz (POINTER) służy do wiązania rekordów w *sety*.

W koncepcji USZBD proponuje się zmiennej długości klucz bazy danych (dorozumiana długość 4 bajty) i o zmiennej strukturze dla różnych obszarów (jakiej — ?). Ponieważ *sety* mogą ze sobą wiązać rekordy z różnych, nawet bardzo odległych obszarów, wobec tego jak można je powiązać dysponując różnymi kluczami bazy danych, stanowiącymi treść odsyłaczy?

2. Autorzy określają jako „dorozumianą" liczbę stron w obszarze na 100, a wielkość strony 3700 bajtów. Moim zdaniem wyrażenie „dorozumiana" wielkość jest to wielkość standardowa, automatycznie określana przez system, gdy projektant nie określi subiektywnej wielkości. W tym przypadku bardzo ważna jest wielkość maksymalna, której autorzy nie podają, a więc można sądzić, że wielkość „dorozumiana" jest jednocześnie wielkością maksymalną. Jeżeli tak, to wielkość obszaru w bazie danych równa zaledwie 100 × 3700 bajtów jest chyba nieporozumieniem.

Obszar w bazie danych jest to logiczny i fizyczny zbiór rekordów o charakterze zbliżonym do tradycyjnego zbioru kartotekowego (FILE). Jest niewątpliwie wskazane, aby rekordy tego samego typu znalazły się w tym samym obszarze bazy danych, tym bardziej, że zakłada się przede wszystkim partiowy sposób przetwarzania bazy danych. Jeżeli rekordem będzie np. PRZEDSIĘBIORSTWO lub PRACOWNIK, to ten typ rekordu będzie miał bardzo wielką liczbę desygnatów (wystąpień wg. terminologii Autorów) i wszystkie one powinny być raczej w jednym obszarze.

W implementacji UNIVAC DMS-1100 określono następujące maksymalne wielkości dla bazy danych:

— liczba obszarów — 4 000
— liczba stron w obszarze — 100 000
— liczba rekordów na stronie — 32 760
— liczba typów rekordów w bazie danych — 4 000
— liczba desygnatów rekordów — nieograniczona
— liczba typów *setów* — 4 000
— liczba desygnatów *setów* — nieograniczona, zależna od liczby desygnatów rekordów macierzystych,
— liczba typów rekordów w *secie* — 63
— liczba desygnatów rekordów w *secie* — nieograniczona

3. Autorzy rezerwują w pamięci zewnętrznej wolną przestrzeń przepełnień, dostępną dla rekordów lokowanych strategią CALC oraz dla celów aktualizacji i modyfikacji, tylko na stronie za pomocą klauzuli LOAD. Przy proponowanej wielkości stron

---

[1]) patrz próba definicji w dalszej części artykułu

jest to absolutnie za mała wolna przestrzeń i dlatego proponuję wprowadzenie stron przepełnień (OVERLAY PAGES) na końcu obszaru oraz przedzielających strony z danymi, podobnie jak w DMS-1100 UNIVAC.

4. Nie rozumiem dlaczego Autorzy rezygnują ze strategii DIRECT. Przecież dla podróży po logicznych związkach w bazie danych, określonych *setami*, konieczne jest oznaczenie logicznego i fizycznego punktu startu. Naturalnie, takim punktem startu może być rekord lokowany strategią CALC, ale z wielu względów nie zawsze jest ona korzystna. Dlatego strategia DIRECT pozwala zapisać w bazie danych rekordy, których główną funkcją jest zapoczątkowanie adresacji innych rekordów lokowanych poprzez *sety*.

5. Autorzy przewidują otwarcie obszarów dla celów aktualizacji i wyszukiwania (UPDATE/RETRIEVAL), a jak jest rozwiązana sprawa początkowego załadowania bazy danych? W DMS-1100 tym celom służy klauzula OPEN FOR INITIAL LOAD, która: a) otwiera obszar, b) powoduje inicjację stron, c) pozwala zapisać rekordy na stronie tylko do granicy określonej klauzulą LOAD, d) automatycznie zapewnia wyłączność *(exclusive)* użycia obszaru, e) upraszcza cały proces przetwarzania w tym przebiegu.

6. Autorzy w zdaniu deklarującym obszar przewidują tylko nazwę obszaru, a nie przewidują żadnego symbolicznego identyfikatora obszaru (np. dla celów określania klucza bazy danych). Tymczasem w zdaniu deklarującym rekord w klauzuli WITHIN nagle pojawia się deklaracja AREA-ID IS, której cel jest zupełnie niezrozumiały.

7. Jeżeli rekord nie mieści się w oznaczonym miejscu i musi być przeniesiony do przestrzeni przepełnień, to autorzy proponują „łamanie" rekordu, przez co rozumieją pozostawienie na właściwym miejscu określonej części rekordu, a do przestrzeni przepełnień przeniesienie pozostałej części rekordu. Może lepiej zastosować w tym celu mechanizm wewnętrznych systemowych odsyłaczy wpisywanych na właściwym miejscu rekordu, a wówczas cały rekrod jest lokowany na stronie przepełnień. To chyba ułatwi przetwarzanie rekordu.

8. W Raporcie CODASYL format identyfikatora bazy danych jest następujący:

data-base-data-name [integer] [{WITHIN / IN} record-name]

Autorzy zrezygnowali z opcyjnego dopisku [integer] zmniejszając elastyczność deklarowania identyfikatora.

9. W zbiorze znaków autorzy znak kreski skośnej '/' określili jako nawias. Funkcja tego znaku jest zupełnie inna niż funkcja nawiasu, który jest znakiem ograniczającym. Kreska skośna może być znakiem dzielenia/łamania, albo może w składni zastępować wyrażenie „lub". Ponieważ autorzy w publikowanej składni nie zastosowali tego znaku, więc jego właściwa funkcja jest niejasna. W każdym razie, dla uniknięcia wieloznaczności, nie można tego znaku nazywać nawiasem. Zresztą w języku angielskim nazywa się to *stroke/slash*.

Może to jeszcze kilka uwag w kwestii terminologii.

Autorzy angielską nazwę SCHEMA i SUBSCHEMA określili jako Schemat i Podschemat. Proponuję polską nazwę Struktura i Podstruktura. Proszę porównać znaczenie terminów: schemat i struktura w słowniku wyrazów obcych.

Dla rekordów wchodzących w skład *setu* proponuję — moim zdaniem — zgrabniejsze nazwy: rekord macierzysty — rekord składowy, zamiast prostego tłumaczenia z języka angielskiego: właściciel — członek. Zamiast wyrażenia „wystąpienie" proponuję termin „desygnat" od dawna używany w logice formalnej dla określenia oznaczonego przedmiotu ze zbioru przedmiotów o tej samej nazwie.

Proponuję natomiast adopcję do języka polskiego angielskiej nazwy SET, ale tylko w kontekście systemu zarządzania bazą danych. Precedensy są liczne, np. komputer, parking itd.

SET jest to zbiór rekordów o określonym kryterium przynależności, uporządkowany wg. takiej relacji, że jeden z rekordów — macierzysty — jest nadrzędny, a wszystkie inne rekordy — składowe — są podporządkowane rekordowi macierzystemu.

W tej mojej, może niezbyt ścisłej definicji, ważniejsza jest cecha relacji porządkującej, niż cecha zbioru, ponieważ SET jest logicznym zbiorem rekordów, natomiast fizycznym zbiorem rekordów jest obszar w bazie danych. Zastosowana przez Autorów dla terminu *set* polska nazwa zbiór jest prostym tłumaczeniem z angielskiego. Biorąc jednak pod uwagę, że w środowisku informatyków słowo ZBIÓR kojarzy się ze zbiorem typu kartotekowego (FILE), może ten termin zaciemniać funkcję obszaru i funkcję *setu* w bazie danych.

Pod adresem Redakcji INFORMATYKI zgłaszam propozycję ostatecznej korekty autorskiej przy trudniejszych tekstach.

Rysunki ilustrujące składnię języka są źle włamane w tekst. I tak, niezbyt szczęśliwie jest włamany rysunek 5, którego część jest na stronie 53, a reszta na stronie 54, a rysunek 7 powinien być na stronie 56. Umieszczenie go na stronie 54 powoduje, że na pierwszy rzut oka zalicza się go do Języka Opisu Danych zamiast do Języka Opisu Pamięci.

Bardzo dużo jest błędów korektorskich i to zarówno w składni języka (rysunki), jak i w tekście. Oto niektóre przykłady:

jest:
LOCUS
CALC
ON DELETE CALC
WITH INTERVAL OF

winno być:
LOCKS
CALL lub odwrotnie
ON DELETE CALL
brak nawiasu opcji

W ten sposób nie wiadomo czasem, czy to jest błąd korektorski, czy wynik proponowanej składni (np. brak TO w klauzuli OCCURS).

Na zakończenie pragnę jeszcze raz podkreślić całą życzliwość dla niezwykle cennej inicjatywy Autorów i moje uwagi są podyktowane li tylko troską, aby Ich ogromne przedsięwzięcie zakończyło się pełnym sukcesem, choćby inny wniosek wynikał z użytych sformułowań, bo taki jest mój sposób bycia.

LITERATURA

[1] Bogucki W. i Staniszkis W. — Koncepcja Uniwersalnego Systemu Zarządzania Bazą Danych dla komputerów JS. INFORMATYKA, Nr 6/75 s. 45.

[2] Bogucki W. i Staniszkis W. — Mechanizmy opisu bazy danych. INFORMATYKA Nr 7—8/75 s. 50.

[3] Bogucki W. i Staniszkis W. — Propozycje Komitetu CODASYL w dziedzinie systemów zarządzania bazą danych. INFORMATYKA, Nr 5/75 s. 37.

Witold Jamontt
OBR SPIS

od autorów

* * *

Jak słusznie zauważył mgr W. Jamontt, Autor „Uwag do koncepcji Uniwersalnego Systemu Zarządzania Bazą Danych dla Komputerów JS" publikacje prezentujące przyjęte przez nas rozwiązania miały na celu spowodowanie dyskusji nad koncepcją systemu.

Pisząc w imieniu całego zespołu realizującego USZBD chcielibyśmy bardzo podziękować mgr W. Jamonttowi za bardzo interesującą próbę oceny przyjętych przez nas rozwiązań.

Niestety, charakter publikowanych w INFORMATYCE artykułów nie pozwala na wystarczająco wyczerpujące omówienie poszczególnych elementów systemu co spowodowało, że Autor „Uwag..." nie zawsze mógł wyciągnąć właściwe wnioski.

Prace nad projektem USZBD były poprzedzone analizą szeregu systemów stanowiących implementację raportów 1969 i 1971 DBTG Komitetu CODASYL (DMS 1100, IDM, DBMS-10, PHOLES, DBMS-DATA SAAB) oraz szeregu publikacji krytycznych.

Niezależnie od implementacji raportów Komitetu CODASYL przeanalizowano również rozwiązanie zastosowane w IMS.

Podstawą realizowanego systemu jest nie tylko raport 1971 DBTG lecz również raport 1973 PDLC *(Data Description Language Committee)* w zakresie Języka Opisu Danych oraz raport 1973 PLC *(Programming Language Committee)* w zakresie Języka Manipulacji Danymi.

W tej sytuacji realizowany system zawiera cały szereg możliwości wynikających z opublikowanych później (po 1971 roku) prac komitetu CODASYL. Trudno jest nie zgodzić się z uwagą, że szerszy zakres realizowanego systemu jest kosztowny. Należy jednak pamiętać, że późniejsze jego modyfikacje wynikające z rozwoju koncepcji Komitetu CODASYL byłyby jeszcze bardziej kosztowne a przewidywany termin realizacji systemu nie usprawiedliwia pominięcia nowych w stosunku do raportu 1971 DBTG możliwości.

Przyjęty sposób realizacji oprogramowania systemu (metoda generacyjna) prowadzi do obciążenia pamięci operacyjnej oprogramowaniem tylko tych funkcji systemu, które są wykorzystywane przez zadanie użytkownika.
Porównywanie koncepcji realizowanego przez nas systemu z rozwiązaniami przyjętymi w systemie DMS 1100 (UNIVAC) jest dosyć ryzykowne bez omówienia ewolucji, jaką przeszła koncepcja DBTG od raportu 1969 (na którym jest oparty DMS 1100) do publikacji z 1973 roku.
Jedną z najważniejszych cech tej ewolucji jest dążenie do uzyskania niezależności oprogramowania użytkowego od struktury bazy danych.
Charakter rozwiązań przyjętych w Języku Manipulacji Danymi służącego do „nawigacji" po strukturze danych nie pozwala na uzyskanie niezależności oprogramowania od struktury danych. Istnieje jednak możliwość pełnego uniezależnienia programów użytkowych od struktury pamięci (fizyczna struktura bazy danych) poprzez wprowadzanie Języka Opisu Pamięci. Wybór struktury pamięci bazy danych jest podyktowany przede wszystkim jej parametrami eksploatacyjnymi oraz możliwościami konfiguracji sprzętu użytkownika.
Język Opisu Pamięci umożliwia sterowanie parametrami eksploatacyjnymi bazy danych (pierwsze deklaracje, późniejsze zmiany) bez konieczności dokonywania zmian w oprogramowaniu użytkowym zastosowań realizowanych w oparciu o tę bazę danych.
Wprowadzenie takiej możliwości jest poważnym krokiem naprzód w stosunku do analizowanych przez nasz zespół systemów zarządzania bazą danych.
Podobnie poważnym usprawnieniem jest wprowadzenie procedur bazy danych umożliwiających rozszerzenie działania opisu bazy danych (Język Opisu Danych) o funkcje proceduralne.
Jednym z typowych zastosowań procedury bazy danych może być automatyczna archiwacja określonych rekordów (niezależna od działania programu użytkowego).
Problem przygotowania odpowiedniej kadry Administratorów Bazy Danych jest rzeczywiście problemem niezwykle poważnym. Ten problem jest normalnym zjawiskiem wprowadzenia każdej nowej technologii (w tym wypadku nowej technologii przetwarzania).
Dysponujemy już w tej chwili obszernym materiałem szkoleniowym i zamierzamy w przyszłym roku prowadzić szkolenia dla pilotowych użytkowników systemu.
Niestety, za zupełnie bezpodstawne musimy uznać stwierdzenie, że ograniczenie się do wersji (zresztą bardzo krytykowanej w literaturze) Języka Opisu Danych DMS 1100 jest zadaniem znacznie trudniejszym, niż wykonanie pełnej wersji Języka Opisu Danych DDLC 1973 i — naszym zdaniem — potrzebnej wersji Języka Opisu Pamięci.
W kontekście przedstawionych uwag w sprawie konkretnych rozwiązań USZAD na JS RIAD odnosi się wrażenie, że ich Autor jest pod silnym wpływem techniki realizacji DMS-1100, który zdaniem wielu specjalistów nie zalicza się do najlepszych tego rodzaju implementacji raportów Komitetu CODASYL. Taka opinia o systemie DMS 1100 jest między innymi spowodowana brakiem mechanizmów ochrony tajności danych w bazie danych *(privacy)*.
W implementacji JS RIAD klucz-bazy danych ma zmienną długość. Jednak dwa klucze-bazy danych o tej samej długości muszą mieć identyczną strukturę. Informacja o budowie klucza-bazy danych znajduje się zawsze w ramach opisu obszaru w schemacie (podschemacie). Za prawidłową interpretację wartości klucza-bazy danych odpowiedzialne jest oprogramowanie obsługi danej wersji struktury pamięci.
Założono, że każdemu obszarowi bazy danych odpowiada dokładnie jeden fizyczny zbiór danych *(data set)* w sensie systemu operacyjnego OS/360. Wymaganie, aby rekordy tego samego typu znajdowały się w tym samym obszarze BD wydaje się być zbyt silnym ograniczeniem. Problem rozmieszczenia typów rekordów w obszarach BD rozstrzyga się na etapie generowania schematu bazy danych. Unikano generalnie ustalania maksymalnych wielkości dla bazy danych poprzez technikę parametryzowania implementacji, pozwalając na zmiany tych parametrów przy generowaniu wymaganej wersji USZBD.
Oto niektóre wielkości maksymalne BD:

- liczba obszarów = maksymalna liczba fizycznych zbiorów danych dla danej instalacji OS/360
- liczba stron w obszarze = nieograniczona
- maksymalna długość strony = pojemność użytkowa ścieżki na dysku
- liczba rekordów na stronie = nieograniczona
- maksymalna liczba typów rekordów = 255.

Przyjęto, że wystąpienia typów rekordów są zmiennej długości. W konsekwencji koniecznością stało się umożliwienie tzw. „łamanie" rekordów na stronach chociażby ze względu na optymalizację wykorzystania pamięci oraz realizacji zasady, że raz przydzielony wystąpieniu rekordu klucz-bazy danych nie może ulec zmianie nawet przy znacznej zmianie długości rekordu. Maksymalna liczba „złamań" nie może jednak być większa niż 15. Nie widzimy potrzeby organizowania specjalnych stron „nadmiarowych" w obszarach, które utrudniałyby tylko zarządzanie zasobami bazy danych.
Celem uzyskania wysokiego stopnia niezależności danych od programów zastosowań autorzy projektu zrezygnowali ze strategii dostępu DIRECT. Uważamy bowiem, że do nawigacji po związkach logicznych w bazie danych wystarczają w zupełności metody CALC i VIA. Stosowanie metody DIRECT zbyt silnie wiąże programy zastosowań ze strukturą danych przechowywanych w pamięci zewnętrznej, co szczególnie uwidacznia się podczas procesów aktualizacji powodujących zmiany w strukturach pamięci BD. Ponadto metoda ta zbyt ogranicza elastyczność zarządzania zasobami bazy danych.
Przydziałem i inicjacją obszarów bazy danych oraz ich wytłumieniem steruje się za pomocą klauzul przyjętego języka opisu pamięci JOP oraz języka *Job Control Language* (ICL/OS/ /360).
Praktycznie odwoływanie się do danych rekordów, setów i obszarów w BD możliwe jest tylko poprzez, wcześniej ustalone nazwy, które mogą się składać maksymalnie z 30 znaków. Autorzy implementacji nie widzą żadnej potrzeby używania dodatkowych identyfikatorów do oznaczania zasobów bazy danych.
W zdaniu deklarującym rekord w klauzuli WITHIN występująca deklaracja AREA-ID IS... daje nazwę zmiennej z programu zastosowań, której zawartość interpretuje się jako nazwa pewnego obszaru BD, opisanego wcześniej w podschemacie.
Schematu/podschematu nie można naszym zdaniem, utożsamiać ze strukturą, podstrukturą ponieważ schemat/ /podschemat może składać się z wielu niespójnych struktur lub podstruktur.
Natomiast godne uwagi są propozycje odnośnie nazewnictwa rekord--właściciel (owner, member) jako rekordu składowego oraz równoważnego stosowania terminów „wystąpienie" i „desygnat". Ponadto popieramy propozycję pozostawiania bez tłumaczenia z angielskiego na język polski terminu set.
Pragniemy również podziękować p. Jamonttowi za uwagi dotyczące błędów korektorskich, które w całości są słuszne i przeprosić za nie Czytelników naszych artykułów.

Witold Staniszkis
Stanisław Wężyk

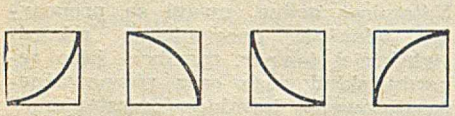
Komputery w plastyce (2)

Od czasu pierwszych ekspozycji komputerowej plastyki w 1965 roku dziedzina ta rozwinęła się na tyle, że jej ogólne omówienie jest zadaniem dość trudnym, wymagającym wstępnej klasyfikacji. Jeśli za kryterium takiego podziału przyjmiemy zasady, na których opierano się przy tworzeniu komputerowych dzieł sztuki, to sprowadzi się to do rozważań nad udziałem przypadku w samym procesie tworzenia. Można wówczas zaproponować podział na cztery grupy, do których należeć będą dzieła powstałe w oparciu o wykorzystanie wyboru:
1) całkowicie przypadkowego
2) przypadkowego, lecz ograniczonego zadanym rozkładem statystycznym
3) przypadkowego, lecz stosowanego jedynie na ograniczonym obszarze
4) przypadkowego, lecz użytego jako środek do znalezienia stanu, który spełniałby założone uprzednio warunki.

Zajmiemy się obecnie omówieniem tych grup, przedstawiając jednocześnie przykłady dzieł, które można by do nich zaliczyć.

* * *

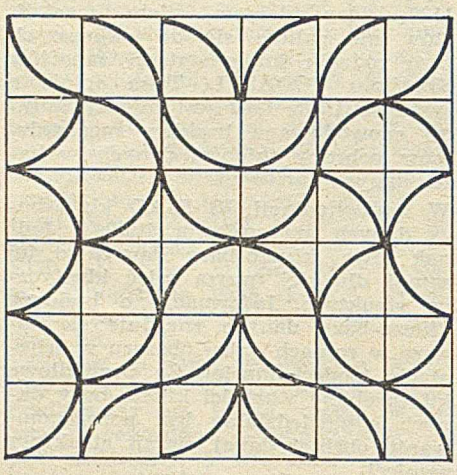
Artysta tworzący obraz na zasadzie wyboru całkowicie przypadkowego musi określić najpierw zbiór, z którego będzie czerpał — zbiór podstawowych elementów graficznych. Np. angielski architekt John Landsdown posługuje się w broszurze „Computer in the Creative Arts" następującym uporządkowanym zbiorem czteroelementowym:



Elementy tego zbioru powstały przez wpisywanie w kwadrat ćwiartki okręgu o środku znajdującym się w jednym z rogów kwadratu. Za pomocą komputera Landsdown losował sześć grup liczb od 1 do 4 po sześć liczb w każdej grupie. Jeśli grupa odpowiadać będzie wierszowi, a liczby z grupy kolumnom obrazu, to otrzymamy obraz będący kwadratem podzielonym na 36 pól (także kwadratów), z których każde zostanie zapełnione przez element podstawowy zgodny z wylosowaną na danej pozycji liczbą. Jeśli więc zdarzy się, że w wyniku losowania maszyna wydrukuje zestaw:

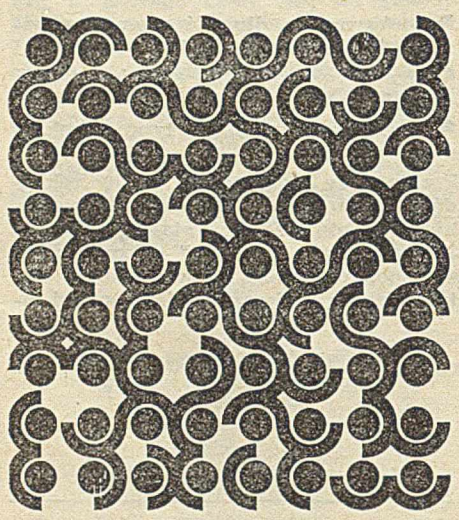
2 4 3 1 2 4
3 3 4 3 1 4
3 2 2 4 1 1
4 1 1 3 2 2
4 4 2 4 2 3
1 2 4 2 4 2

to odpowiadać mu będzie kompozycja plastyczna:



Artysta komputerowy Colin Sheffield stosuje tylko jeden element podstawowy. Jest nim czarne koło z półpierścieniem:

A oto jeden z rezultatów uzyskany w 1970 roku przez Sheffielda, który polecił maszynie dokonanie przypadkowego zestawiania elementu podstawowego obracanego w czterech kierunkach.



Zupełnie inne elementy podstawowe stosuje R. J. Botting z Brunel University.

Przyjmuje on, że zbiór elementów złożony jest z linii prostych o długości będących wielokrotnością odcinka uznanego za jednostkę miary. Liczby od 0 do 9 określają ilość takich jednostkowych odcinków w danej prostej. Z gramatyką związane są liczby innej serii. Wylosowanie 1 oznacza przesunięcie następnej linii o 30° w lewo w stosunku do poprzedniej; 2 przesunięcie o 20° w lewo; 3 o 10° w lewo; 4, 5 i 6 kolejno o 10°, 20° i 30° w prawo. Liczba 7 powoduje zakończenie danej gałęzi i budowanie drzewa począwszy od ostatniego rozgałęzienia, 8 i 9 to samo, ale o dwa i trzy rozgałęzienia w tył. 0 w tej serii kończy budowanie drzewa.

O pozostałych grupach plastyki komputerowej pisać będziemy za miesiąc. Na zakończenie chcielibyśmy przedstawić kompozycję będącą jednym z bardziej zaawansowanych i atrakcyjnych przykładów grafiki plotterowej.

Marek Hołyński

Recenzja książki E. I. Lowe'a, A. E. Hiddena: Maszyny cyfrowe w automatyce przemysłowej*)

Chociaż maszyny cyfrowe stosuje się w sterowaniu różnego rodzaju procesami np. technologicznymi, już od kilkunastu lat, to na nadmiar publikacji zarówno o charakterze ogólnym jak i traktujących o konkretnych rozwiązaniach nie można narzekać. Spowodowane jest to między innymi pewną niechęcią do publicznego ujawniania szczegółów technicznych w tym zakresie.

Istniejące zaś prace, np. [2, 3, 5, 9] — zawierają zazwyczaj wyniki teoretycznych rozważań dotyczących w większości przypadków algorytmów sterowania procesami (często optymalnego). Wprowadzony w nich formalizm matematyczny odstrasza praktyków, dla których zresztą budowa algorytmów jest tylko etapem w projektowaniu i wdrażaniu systemów sterowania.

I właśnie z myślą przede wszystkim o praktykach, o ludziach, których praca nie kończy się na papierze lecz w hali produkcyjnej nakładem Wydawnictw Naukowo-Technicznych w Warszawie ukazało się tłumaczenie książki dwóch angielskich autorów E. I. Lowe'a i A. E. Hidden'a pt. Maszyny cyfrowe w automatyce przemysłowej.

Autorzy podjęli próbę przedstawienia niespecjalistom, praktykom całokształtu problematyki sterowania procesów technologicznych z użyciem maszyny cyfrowej [1]).

Z systemami sterowania procesów z użyciem maszyny cyfrowej wiążą się następujące problemy:
— wnikliwe poznanie procesu, przebiegiem którego sterować ma maszyna cyfrowa
— budowa i działanie maszyny cyfrowej, urządzenia peryferyjne, oprogramowanie
— układy automatyki, regulacji
— opłacalność sterowania z maszyną cyfrową
— miejsce maszyny cyfrowej w systemie sterowania
— funkcje maszyny cyfrowej sterującej procesem
— połączenie operator-maszyna-proces
— metody projektowania systemów sterowania (w szczególności sterowania optymalnego) z maszyną cyfrową
— sposoby i kryteria doboru maszyn cyfrowych oraz zestawu urządzeń peryferyjnych do sterowania konkretnymi procesami
— organizacja projektowania systemów oraz prac wdrożeniowych.

Jest rzeczą oczywistą, że chcąc rozważyć wszystkie wspomniane problemy w rozsądnej objętości i w sposób podstawowy rezygnując z pomocy matematyki należy przyjąć pewien poziom ogólności rozważań. Trafne zatem wydaje się być określenie zawarte w przedmowie do wydania polskiego, że książka prezentuje filozofię zastosowań maszyn cyfrowych w systemach sterowania.

Układ tematyczny książki umożliwia Czytelnikowi poznanie dostatecznie szerokiego wachlarza problemów sterowania procesów z maszyną cyfrową — począwszy od rozważań nad istotą procesu sterowanego, a skończywszy na zagadnieniach organizacji prac wdrożeniowych.

W pierwszych trzech rozdziałach omawia się kolejno proces produkcyjny, problem sterowania oraz wykorzystania w tym celu maszyn cyfrowych. Autorzy wprowadzają i wyjaśniają pojęcie hierarchii procesów, a następnie prezentują naturalny, hierarchiczny „model" systemu zarządzania i sterowania. Na przykładzie kierownika (jednostki nadrzędnej) i operatora (jednostki podrzędnej) pokazany jest przepływ informacji w systemie z uwzględnieniem interakcji pomiędzy poszczególnymi jednostkami decyzyjnymi. Pominięto tu jedynie warunki koordynacji i zgodności problemów decyzyjnych na kolejnych szczeblach hierarchii, których spełnienie jest niezbędne dla funkcjonowania systemu. Bliższe szczegóły na ten temat wraz z przykładami można znaleźć np. w [2, 4].

W sposób bardzo elementarny autorzy definiują także funkcję zysku i jej składniki.

Rozdział drugi poświęcony jest omówieniu najniższego poziomu hierarchii zarządzania, gdzie decyzje lub ich skutki ujawniają się jako sterowanie bezpośrednie, czyli bezpośrednie oddziaływanie na proces w celu wywarcia wpływu na jego zachowanie w zgóry określony sposób. Podaje się proste przykłady systemów sterowania w układach otwartych i zamkniętych przy zakłóceniach. Zwraca się także uwagę na możliwość sterowania nadrzędnego za pomocą środków automatyki konwencjonalnej.

Autorzy sygnalizują tu możliwość użycia maszyny cyfrowej do sterowania zarówno bezpośredniego jak i nadrzędnego.

W dalszej części omawiana jest ogólna budowa i zasady działania maszyny cyfrowej. Autorzy prezentują tu także wielce pouczający schemat połączeń maszyny z procesem, wskazując na fakt istnienia różnic pomiędzy maszyną stosowaną „w linii" i stanowiącą główny temat tej książki, a maszyną przeznaczoną do innych celów, np. obliczeń inżynierskich.

Rozdział czwarty poświęcony jest w całości zagadnieniom najistotniejszym dla projektanta systemów sterowania z maszyną cyfrową tj. sterowaniu bezpośredniemu i nadrzędnemu z maszyną cyfrową oraz opłacalności i efektom wynikającym z zastosowania maszyny cyfrowej. Bardzo przejrzyste schematy ilustrują wykorzystanie maszyny cyfrowej na różnych szczeblach hierarchii zarządzania i sterowania. Zbyt mało miejsca poświęcili jednak autorzy sprawom kosztów sterowania cyfrowego. Wydaje się, że wykres zależności kosztów od liczby układów regulacji, pochodzący z połowy lat sześćdziesiątych, jest nieaktualny jako kryterium porównawcze systemów: konwencjonalnego i cyfrowego. Przy aktualnych kosztach systemów sterowania z maszyną cyfrową oraz sprzętu analogowego sprawa kosztów porównawczych nie powinna być istotnym argumentem przemawiającym za stosowaniem maszyn cy-

*) WNT-Warszawa 1975, str. 259, ark. wyd. 23,2

[1]) Termin „maszyna cyfrowa" używany jest zgodnie z terminologią przyjętą w recenzowanej książce.

frowych. W konsekwencji problem nie jest do końca wyjaśniony.

W rozdziale piątym poruszane są kolejno trzy odrębne lecz wzajemnie ze sobą powiązane tematy: sprzęt, oprogramowanie i otoczenie maszyny cyfrowej, istotne zwłaszcza przy wyborze i instalowaniu maszyny cyfrowej do sterowania. Nie podaje się jednak żadnych kryteriów wyboru i metod, koncentrując się głównie na werbalnym opisie.

Problemy porównywania maszyn cyfrowych w oparciu o tzw. mieszanki statystyczne znaleźć można np. w [6, 8]. Czytelnika pragnącego rozszerzyć wiadomości rozdziałów 3 i 5 odsyłam do dostępnych na półkach księgarskich pozycji [1, 7].

Kolejny rozdział poświęcony jest omówieniu urządzeń wejściowych i wyjściowych odgrywających istotną rolę w automatyzacji procesów technologicznych. Przedyskutowano tu problemy łączności między maszynami cyfrowymi w obrębie hierarchicznego systemu sterowania, łączności z obsługą procesu oraz cyfrowych i analogowych wejść i wyjść procesowych maszyny cyfrowej.

Przedmiotem rozważań podziału źródłowego są przyrządy pomiarowe przekazujące sygnały z procesu do urządzeń wejściowych maszyny cyfrowej oraz sprzęt łączący jej urządzenia wyjściowe z procesem. Między innymi podaje się wymagania w zakresie dokładności oraz precyzji tych przyrządów. Omówiono także częściej stosowane pomiary. Autorzy mocno podkreślili wpływ zakłóceń elektrycznych na zniekształcenie sygnałów przesyłanych pomiędzy procesem i maszyną wyjaśniając także ich charakter i źródła. Rozdział ósmy wprowadza czytelnika w sposób w podstawy matematyczne i strategie sterowania. Autorzy poruszają tu problemy próbkowania i filtrowania sygnałów, sterowania optymalnego, teorii procesów stochastycznych. Trafnie i interesująco opisano zagadnienie optymalizacji dynamicznej. Szkoda tylko, że autorzy pominęli takie zagadnienia jak np. układy uczące się, metody Monte Carlo, symulację cyfrową, nie mniej przydatne przy projektowaniu automatyzacji procesów. W rozdziale poświęconym ekonomice sterowania z maszyną cyfrową, autorzy starają ustosunkować się do zagadnień kosztów oraz korzyści wynikających z użycia maszyny cyfrowej. Nie starają się jednak precyzować zależności np. procentowego udziału kosztów maszyny cyfrowej, oprogramowania i oprzyrządowania wraz z montażem i uruchomieniem systemu w kosztach ogólnych autofnatyzacji, wyjaśniają to dopiero przypisy tłumacza. Przykłady „udanych" systemów sterowania z maszyną cyfrową są ubogie, można je w szerszym ujęciu znaleźć w np. [1].

Rozdziały dziesiąty i jedenasty obejmują zagadnienia organizacji projektowania systemów oraz prac wdrożeniowych. Przywiązując duże znaczenie do takich problemów jak studia szczegółowe, dobór zespołu projektującego, planowanie, organizacja i instalacja systemu, uruchamianie systemu, jego eksploatacja i konserwacja, autorzy podają cenne wskazówki, które mogą stanowić pożyteczną pomoc w samodzielnym postępowaniu.

W rozdziale dwunastym omawia się perspektywy rozwojowe techniki maszyn cyfrowych oraz sterowania procesów z ich zastosowaniem.

Rozdział trzynasty ma charakter dodatku i jest chyba pewnym nieporozumieniem, ponieważ zawiera stosunkowo zaawansowany formalizm matematyczny, którego uniknięcie było jednym z zasadniczych celów autorów. Dla porządku podaję, że zawiera on wyprowadzenie algorytmu sterowania bezpośredniego dla regulatora PID oraz charakterystykę metod wektorowych i zmiennych stanu procesu. Książka zaopatrzona jest w słowniczek wybranych terminów co znacznie ułatwia niefachowemu czytelnikowi zrozumienie jej treści.

Sądzę, że przytoczona powyżej charakterystyka książki zobrazuje czytelnikowi ogrom problemów związanych ze stosowaniem maszyn cyfrowych do sterowania procesami. Autorzy w sposób bardziej lub mniej wyczerpujący omówili problemy, które zasygnalizowano na początku recenzji. Nie ustosunkowali się tylko do metod projektowania wielkich cyfrowych systemów sterowania. Jest rzeczą wiadomą, że zaprezentowany opisowo aparat matematyczny w przypadkach złożonych okazuje się niewystarczający. Istnieją jednak inne efektywne metody projektowania np. symulacyjne.

Powyższa uwaga nie umniejsza niewątpliwych walorów tej książki, do których zaliczyć można przede wszystkim — niezwykłą przystępność i komunikatywność pozwalające łatwo przyswoić niespecjalistom poruszane tu, często trudne problemy
— interesujące „filozoficzne" podejście, umożliwiające objęcie całokształtu problematyki
— nacisk na stronę praktyczną
— ujednoznacznienie i objaśnienie terminologii
— logikę ujęcia osiągniętą przez właściwy układ sekwencji rozdziałów.

Istotnym walorem książki jest także jej bardzo dobre tłumaczenie, w którym widać próbę ujednolicenia terminologii polskiej w tej dziedzinie. Obszerne przypisy wyjaśniają wszelkie nieścisłości, a często rozszerzają materiał zawarty w książce, umożliwiając jej lepsze zrozumienie.

Nie znajdzie tu Czytelnik szczegółów, konkretnych rozwiązań systemów sterowania z użyciem maszyny cyfrowej. Otrzyma jednak porcję wiedzy, która wprowadzi go w problematykę sterowania cyfrowego wielkich systemów. Pobudzona zostanie jego wyobraźnia, a inspirujący charakter tej książki może zjednać wspomnianym zagadnieniom całą rzeszę zwolenników.

Z tego względu polecam tę książkę nie tylko praktykom niespecjalistom, kadrze kierowniczej w zakładach przemysłowych, studentom, ale również wszystkim tym, którzy do tej pory nieufnie spoglądają na zjawisko inwazji informatyki do przemysłu.

Zbigniew Poznański

LITERATURA

[1] Eadie D.: Nowoczesne maszyny i systemy cyfrowe Seria: *Informatyka*, WNT, Warszawa 1975.
[2] Findeisen W.: Wielopoziomowe układy sterowania PWN, W-wa 1974.
[3] Kulikowski R.: Sterowanie w wielkich systemach WNT, Warszawa 1974.
[4] Mesarovic M. D., Macko D., Takahara Y.: Theory of hierarchical multilevel systems. ACADEMIC PRESS, N.Y. 1970.
[5] Styczeń K.: System transportowy w dziale przygotowania surowca w cementowni, model matematyczny i algorytm sterowania. PODSTAWY STEROWANIA T. 4, zeszyt 4, PWN 1974.
[6] Targowski A.: Porównywanie komputerów metodą Gibsona. Mat. Symp. nt. Metody komparastyczne techniki obliczeniowej W-wa 1969.
[7] Turski W. M.: Propedeutyka informatyki. PWN, W-wa 1975.
[8] Podstawy użytkowania maszyn cyfrowych w ośrodkach naukowo-badawczych. PWN, W-wa 1973.
[9] Praca zbiorowa: Teoria i zastosowanie wielkich systemów, OSSOLINEUM 1972.

ZAKŁAD ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ
w Koszalinie
posiada do odstąpienia
dwa zestawy tabulatorów
„SOEMTRON 402"
z dziurkarką sumaryczną
„SOEMTRON 441"
i sorterem
„SOEMTRON 434"
oraz zestaw
EKD „ODRA 1103"
z reproducerem
PR-80-2.
Zgłoszenia prosimy kierować do ZETO
w Koszalinie, ul. Pawła Findera 93a,
tel. 271-81.

WCT/1704/K/75

dokończenie ze str. 22

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Testy pamięci zewnętrznych m.c. R-30. Testy autonomiczne. Z.1. C40007-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 47.
Testy: taśm magnetycznych (KNTP), pamięci dyskowej (KNTP), interface'u pamięci dyskowej (KNTP), powierzchni pamięci dyskowej (KNTP).
Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Testy jednostki centralnej m.c. R-30. Testy autonomiczne. Z.1. C40002-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 73.
BCR2 bazowy test procesora. Testy: pamięci operacyjnej, procesora, kanału multipleksora, kanałów selektorowych, współpracy kanałów, pamięci lokalnej, kluczy ochrony pamięci, pamięci operacyjnej (KNTP).
Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Programy organizacyjne m.c. R-30. Opis programu FOFF. Z.3. C40017-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss. 56.
Program kompilujący i aktualizujący bibliotekę testów: funkcja, wymagania programu, przedmiot sprawozdania, instrukcje operatorskie, komunikaty, sygnalizacja błędów.
Materiały przeznaczone są dla operatorów i programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie. Testy jednostki centralnej m.c. R-30. Test — sekcje DMES. Z.2. C40003-1.** Wyd. MERA-ELWRO, Wrocław 1975, ss .80.
Testy: jednostki arytmetyczno-logicznej, licznika, rozkazów, specjalnych sekwencji rozkazów, przerwań programowych, jednoczesny pracy AL i kanału SEL, funkcji rozkazu diagnoza rozkazu (DIAG 1), funkcji diagnoza rozkazu (DIAG 2), funkcjonalny czasomierza, pamięci operacyjnej, pamięci lokalnej, pamięci roboczej SSK-MPX, kluczy ochrony pamięci operacyjnej, kanału multipleksora, kanału selektora.
Materiały przeznaczone są dla operatorów i programistów.

● **Maszyny cyfrowe i ich zastosowanie — HELLWIG 2.** — red. PWE, Warszawa 1975, ss. 472, cena zł 65.—
Wstęp: historia, podział, organizacja i zastosowanie maszyn cyfrowych. Arytmetyczne podstawy maszyn cyfrowych. Algorytmy i schematy blokowe. Język ALGOL 60. Język COBOL. Programowanie w języku PLAN. Techniki projektowania systemów informacyjnych. Procedury i programy podstawowych zagadnień statystyki i ekonometrii.
Książka wprowadzająca przeznaczona jest dla ekonomistów oraz programistów zagadnień ekonomicznych.

● **Logiczne tablice decyzji. Podstawy teoretyczne i zastosowania praktyczne — NIEDŹWIECKI.** Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1975, ss. 191, cena zł 58.—
Techniki decyzyjne. Teoretyczne podstawy budowy i zastosowań logicznych tablic decyzji. Struktura logicznych tablic decyzji. Układanie logicznych tablic decyzji. Podstawy podziału złożonego problemu w szeregu sprzężonych logicznych tablic decyzji. Przekształcanie logicznych tablic decyzji na programy komputerowe. Stosowanie logicznych tablic decyzji w systemach automatycznego przetwarzania informacji. Stosunek nazw.
Materiały przeznaczone są dla projektantów systemów EPD oraz programistów.

● **Pamięci półprzewodnikowe scalone — EIMBINDER,** red.. Tłum. wyd. ang. z roku 1971. WNT, Warszawa 1975, ss. 209, cena zł 29.—
Układy scalone stosowane w pamięciach półprzewodnikowych. Analiza zastosowań pamięci typu odczyt/zapis. Wytwarzać czy kupować pamięci półprzewodnikowe. Pamięci półprzewodnikowe; logika prądowa na tle innych rozwiązań. Pamięci stale MOS. Obszary zastosowania pamięci MOS. Pamięci skojarzeniowe MOS. Układy scalone MOS o symetrii komplementarnej. Własności układów scalonych MOS o symetrii komplementarnej. Obszary zastosowań układów MOS. Azotkowe układy scalone MNS. Generacja znaków przy użyciu pamięci MOS. Zastosowanie układów MOS do mikroprogramowania. Pamięci operacyjne MOS. Pamięci bipolarne o dostępie swobodnym. Projektowanie urządzeń z pamięciami bipolarnymi. Definicje dotyczące układów o średniej skali integracji. Konwersja liczb binarnych na dziesiętne kodowane binarnie. Analiza kosztów i niezawodności pamięci z doprowadzeniami belkowymi.
Książka przeznaczona jest dla inżynierów elektroników specjalizujących się w pamięciach półprzewodnikowych.

● **Maszyny cyfrowe w automatyce przemysłowej — LOWE E. I., HIDDEN A. E.** Tłum. wyd. ang. z 1971 r. WNT, Warszawa 1975, ss. 259, cena zł 45.—
Procesy i zarządzanie. Sterowanie bezpośrednie. Maszyna cyfrowa. Maszyna cyfrowa w sterowaniu procesów. Maszyna cyfrowa: szczegóły. Urządzenia wejściowe i wyjściowe. Oprzyrządowanie systemów sterowania. Podstawy matematyczne i strategia dla sterowania z maszyną cyfrową, wprowadzenie opisowe. Ekonomika sterowania z maszynami cyfrowymi. Organizacja projektowania systemów sterowania. Podstawy matematyczne i strategie dla stespektywy. Dodatek. Słownik.
Książka zapoznająca z podstawami automatyzacji kompleksowej przeznaczona jest dla kadry kierowniczej zakładów przemysłowych oraz dla automatyków i technologów praktyków.

● **Tablice decyzji — POLLACK S. L., HICKS H. T., HARRISON W. J.** PWN, Warszawa 1975, ss. 275, cena zł 40.—
Rys historyczny. Struktura tablic decyzyjnych. Rozwinięcie teorii tablic decyzyjnych. Twierdzenia teorii tablic decyzyjnych. Warunki: Kryteria decyzji. Czynności. Optymalizacja tablic decyzyjnych. Reguła: Nadmiar i sprzeczność. Reguła: Kompletność i łączenie reguł. Języki używane w tablicach decyzyjnych. Analiza systemu. Tablice decyzyjne w programach dla maszyn cyfrowych. Pisanie tablicy decyzyjnej. Poprawianie programów zawierających tablice decyzyjne. Wdrażanie tablic decyzyjnych. Wzajemne oddziaływanie tablic decyzyjnych. Studium przykładowe: Budowa i rozkład tablicy decyzyjnej. Algorytmy rozkładu. Dodatki: Dowody twierdzeń o tablicach decyzyjnych. Wprowadzenie do COBOL-u. Algorytmy tłumaczenia tablic decyzyjnych.
Książka przeznaczona jest dla projektantów systemów EPD oraz programistów.

● **Biblioteki programów i podprogramów maszyn ODRA serii 1300 — STANKO J.** Wyd. Politechniki Wrocławskiej, Wrocław 1975, ss. 242, cena zł 22.—, Biblioteka WASC.
Ogólne wiadomości o rodzajach oprogramowania i zasadach przechowywania. Organizacja, tworzenie i aktualizacja bibliotek programów i podprogramów na nośnikach magnetycznych. Biblioteki programów i podprogramów ogólnego użytku dostępne dla użytkowników.
Opisy programów tworzenia i aktualizacji, własnych bibliotek programów i podprogramów użytkowych.
Opracowanie może być użyteczne dla projektantów systemów, programistów, inżynierów, operatorów maszyn cyfrowych, archiwistów, studentów oraz użytkowników WASC.

oprac. **Jerzy Klamborowski**

ORWO — Taśmy magnetyczne ORWO — partnerem EPD

Wysokiej jakości taśmy magnetyczne przyczyniają się w coraz większym stopniu do opanowania procesów informatyki w krajach wspólnoty socjalistycznej.
Powstała w stosunkowo krótkim czasie Fabryka Taśmy Magnetycznej — VEB Dessau, jest tym zakładem Kombinatu Fotochemicznego NRD, który specjalizuje się w produkcji taśmy magnetycznej używanej do rejestrowania danych cyfrowych.

Naszą taśmę magnetyczną produkują nowoczesne, wysoko wydajne urządzenia automatyczne.

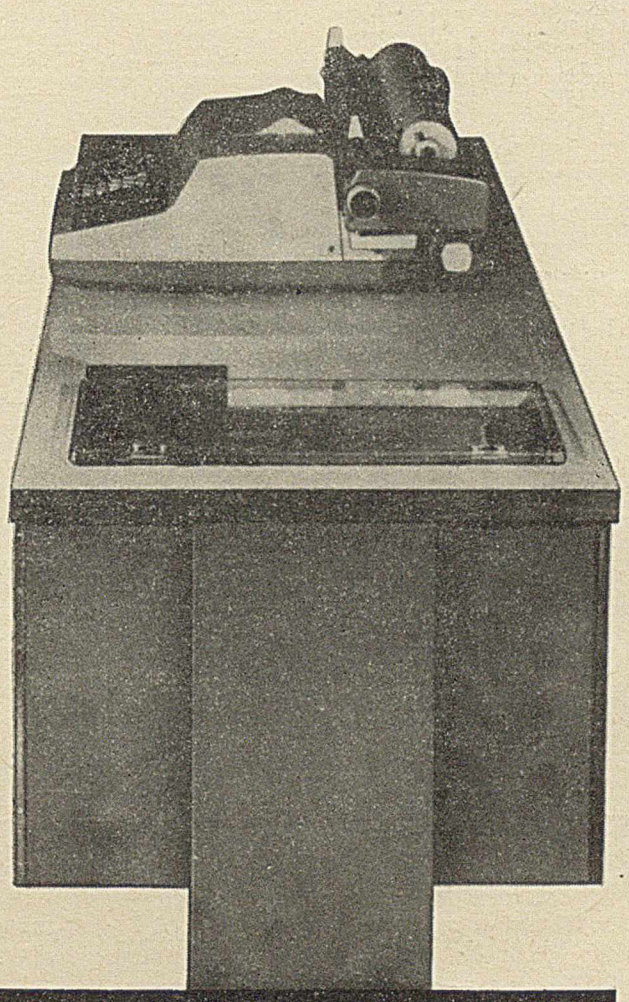
VEB Magnetbandfabrik Dessau,
zakład Kombinatu VEB Filmfabrik Wolfen
Eksporter:
Kamera-Film Export-Import
Volkseigener Aussenhandelsbetrieb der DDR
1055 Berlin/DDR
Storkower Strasse 120

WCT/1500/K/75

Zanim zagłębicie się w problemy ewidencjonowania danych: daro 1333

daro 1333

Ewidencjonowanie danych może stać się wąskim gardłem. Kto może temu zapobiec?
Na przykład nasze urządzenie daro 1333.
Gdy informacje będą rejestrowane na taśmie perforowanej, a następnie przetwarzane i analizowane na komputerach międzyzakładowych lub wewnątrzzakładowych powstaje oczywiście wiele zagadnień. Zarówno pionowe, jak i poziome zapamiętywanie wartości. Saldowanie. Przenoszenie, wyznaczanie kontrolnych sum częściowych. Kontrola techniczna...
Zainteresujcie się tym racjonalnym systemem, zanim u Was powstaną problemy ewidencjonowania danych.
Nasza dokumentacja pozostaje do Waszej dyspozycji. Prosimy napisać do nas.



bme

Büromaschinen-Export
GmbH Berlin
DDR — 108 Berlin,
Friedrichstrasse 61
Niemiecka Republika
Demokratyczna

Przedstawicielstwo BME
w Polsce
Biuro Techniczno-Handlowe przy Ambasadzie
NRD
Warszawa, ul. Filtrowa
62 m. 63

Sprzedaż i informacje:
Predom-Org.,
00-033 Warszawa
ul. Górskiego 9

WCT/1365/K/75

benson

francuski producent urządzeń infograficznych

Francuska firma BENSON S.A., jedyny europejski producent pełnej gamy peryferyjnych urządzeń infograficznych do maszyn cyfrowych oferuje:

— autokreślarki (plottery) — modele TD 210, 222, 232 płaskie oraz TD 111, 121, 122, 132 bębnowe, charakteryzujące się wysoką precyzją wykonania i działania, dużymi szybkościami rysowania oraz możliwością programowego wyboru pióra, służące do otrzymywania precyzyjnych rysunków i wykresów z komputera.

Podstawowe dane techniczne autokreślarek

111	112	121	122	132	Charakterystyka	210	222	232
34 50	34 50	74 50	74 50	94 50	cm Wymiary rysunku y cm cm x cm	75 48	120 84	150 84
1	3	1	3	3	Liczba pisaków	1	4	4
	×		×	×	Wbudowany interpolator		×	×
0,1 14	0,1 21	0,1 12,7	0,1 21	0,1 21	Krok programowy mm Maks. prędkość kreślenia cm/s	0,1 8,5	0,05 10,6	0,05 10,6

— czytniki punktów i krzywych LP 620, LC 610, LF 630 do wczytywania (digitalizacji) informacji graficznych z rysunków, map, mikrofilmów do komputera

— drukarki i autokreślarki (plottery) mikrofilmowe COM 310, 320, 330 rejestrujące na mikrofilmie teksty alfanumeryczne i rysunki z komputera z szybkościami do 15 000 linii/min (drukarka) lub 200 000 kroków/s (autokreślarka) przy zdolności rozdzielczej obrazu 6484 × 8192 punktów i 8 poziomach jasności plamki świetlnej wybieranych programowo.

MASZYNY INFOGRAFICZNE BENSONA MOGĄ PRACOWAĆ ON-LINE LUB OFF-LINE ZE WSZYSTKIMI ZNANYMI KOMPUTERAMI I MINIKOMPUTERAMI (W TYM ODRA 1200 I 1300, MIŃSK 32 I KOMPUTERY SERII RIAD) I DOSTARCZANE SĄ WRAZ Z OPROGRAMOWANIEM.

DO STEROWANIA OFF-LINE AUTOKREŚLARKAMI I DRUKARKAMI COM BENSON OFERUJE MODUŁOWE CZYTNIKI TAŚMY MAGNETYCZNEJ SERII 400 (7 I 9 ŚCIEŻEK, 556, 800, 1600 BPI).

Maszyny infograficzne BENSONA pracują już w Warszawie, Krakowie, Bydgoszczy, Toruniu, Sosnowcu, Poznaniu i Gdańsku. Stały serwis techniczny w Polsce zapewniony.

Katalogi, szczegółowa dokumentacja techniczna, informacje, oferty
TRANCHANT ELECTRONIQUE BIURO TECHNICZNE W WARSZAWIE
00-020 Warszawa, ul. Szpitalna 1 m. 36, telefon 27-56-40, 26-80-31 w. 35 i 36, teleks: Tranel 81 46 43 pl.

BENSON S.A., 1, rue Jean Lemoine, Z.I. Petites Haies, 94 — Créteil FRANCE
telefon 899-10-90, teleks: Bensonsa 21845 f.