



P. 1877/79

3

1979

informatyka

Granice i instrumenty spójności centralnych systemów informatycznych. Część 2 <i>Józef Oleński</i>	1
Możliwości automatycznej korekcji błędów składniowych w programach <i>Zbigniew Kierzkowski, Jacek Koperski</i>	3
Czynniki wpływające na metodykę projektowania systemów informatycznych <i>Zdzisław Zapolski</i>	6
Wyszukiwanie i sortowanie danych za pomocą układu komórkowego <i>Mieczysław Muraszkiewicz</i>	9
Komputeryzacja gospodarki materiałowej na przykładzie województwa wrocławskiego <i>Jerzy Szczucki</i>	12
Między UNIDATĄ a HONEYWELLEM. Część 1 <i>Piotr Strzałkowski</i>	14
Mierzenie jakości i wydajności programowania. Część 1 <i>Ewa Józwiak</i>	17
SZTUCZNA INTELIGENCJA	
Na trzy głosy Nie ma jednolitej definicji — <i>Andrzej Dziurnikowski</i>	19
Mizerne naśladownictwo — <i>Marek Gliński</i>	20
Heurystyka pomoże — <i>Agnieszka Szewczyk</i>	20
ZE ZJEDNOCZENIA INFORMATYKI	
Za i przeciw systemom powtarzalnym <i>Artur Hajnicz</i>	21
Krajowa Konferencja DIEBOLDA <i>Władysław Klepacz</i>	24
ÓŚRODKI INFORMATYKI PREZENTUJĄ	
Programowanie modularne w praktyce <i>Jiří Zaveský</i>	25
Z KRAJU	
Wedle stawu grobla <i>Krystyn Bernatowicz</i>	26
Czy „branżowy” stanie się „centralnym”? <i>Krystyn Bernatowicz</i>	27
Zastosowanie komputerów w przemyśle	29
ZE ŚWIATA	
„Należy tworzyć swoje własne systemy...” IFIP 80	30 31
NASZE RECENZJE	
Narodziny nowego działu statystyki gospodarczej <i>Adam B. Empacher</i>	34
MERA-ELWRO	
Oprogramowanie komputerów JS Urządzenia pomocnicze dla ośrodków obliczeniowych Zmiany konstrukcyjne w MTS 304-2 Umowa serwisowa MERA-ELWRO—MERA-ELZAB Kto może być przyjęty na kursy? Działalność Rady Kompleksowej Obsługi JS EMC <i>Oprac. Jerzy Jankowski</i>	36
PROBLEMATYKA BAZY DANYCH	
Bank danych statystycznych w technologii DBTG CODASYL <i>Marek Lasota</i>	37
ZAGADNIENIA PRAWNE	
Patentowanie oprogramowania komputerów	III str. 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), mgr Janusz GWIAZDA, dr inż. Marek HOLYŃSKI, mgr inż. Stanisław
JASKÓLSKI, Władysław KLEPACZ (zast. redaktora naczelnego), mgr Stanisław MROZIK,
dr inż. Tomasz PAWLAK. Sekretarz redakcji: Anna GLUTH-NOWOWIEJSKA
Red. techn.: EWA KAMIŃSKA

RADA PROGRAMOWA

Prof. dr hab. Tadeusz PECHE (przewodniczący), mgr inż. Tomasz BAŃKOWSKI (sekretarz),
mgr inż. Antoni BOSSOWSKI, mgr inż. Roman BURNO, prof. dr hab. Andrzej JANICKI,
mgr inż. Jan KRAMARCZUK, prof. dr hab. inż. Juliusz KULIKOWSKI, prof. dr hab. Leon
ŁUKASZEWICZ, gen. dr inż. Marian PASTERNAK, mgr inż. Bronisław PIWOWAR, mgr
Zbigniew SUBSTYK, mgr Jerzy TRYBULSKI, doc. dr hab. Tadeusz WALCZAK, dr inż.
Jan ŻYDOWO

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pokój 331, tel. 27-71-40 lub centrala 26-82-61 w. 285, dyżury redakcji 10.00—13.00

Zakł. Graf. „Tamka”. Zam. 52. Papier druk. sat. IV kl. 70 g. A1. Obj. 5 ark. druk. Nakład 6750. C-102.

JÓZEF OLEŃSKI

Ośrodek Badawczo-Rozwojowy Systemu
Państwowej Informacji Statystycznej przy GUS
Warszawa

Granice i instrumenty spójności centralnych systemów informatycznych. Część 2

W części 1 artykułu (INFORMATYKA nr 2/79) przedstawiono podstawowe problemy spójności centralnych systemów informatycznych oraz sformułowano postulaty odnośnie do granic spójności.

Praktyczna realizacja tych postulatów uwarunkowana jest odpowiednimi instrumentami, pozostającymi do dyspozycji administracji centralnej i projektantów systemów, a zapewniającymi możliwość oddziaływania na systemy. Z instrumentów wymienionych w części 1 artykułu na szczególną uwagę zasługują:

- wspólna baza pojęciowa CSI
- klasyfikacje, nomenklatury, języki informacyjne
- wspólne podsystemy instrumentalne
- organizacja służb informacyjno-koordynacyjnych.

Wspólna baza pojęciowa centralnych systemów informatycznych jest bazowym instrumentem spójności informacyjnej i językowej.

W Głównym Urzędzie Statystycznym — przy współpracy z Komisją Planowania przy RM, resortami i jednostkami — prowadzi się prace nad definicjami pojęć występujących w planowaniu i statystyce. Celem tych prac jest zapewnienie przejścia między kategoriami „konceptualnymi”, jakimi operuje w swojej praktyce planista, analityk i badacz, a kategoriami ewidencyjnymi, jakie występują w systemach informacji statystycznej i planistycznej. Rozwój metod planowania i zarządzania powoduje względnie szybkie starzenie się niektórych pojęć i definicji. Realizacji licznych postulatów zmierzających do przyspieszenia tych prac powinno towarzyszyć unowocześnienie formy udostępniania ich wyników. Dotychczasowa forma (publikacje w Zeszytach Metodologicznych GUS) powinna być wspomaganą serwisem informacyjnym o odpowiedniej selektywności, ewentualnie z wykorzystaniem techniki komputerowej.

Należy podkreślić, że szczególnie ważną przy definiowaniu pojęć jest centralna koordynacja prac. Lokalne ini-

cyjatyw, wynikające często z autentycznej potrzeby, prowadzone bez niezbędnej koordynacji z pracami centralnie sterowanymi przez Główny Urząd Statystyczny, mogą wprowadzać niepotrzebny i szkodliwy zamęt w rozumieniu tych samych pojęć. Drobne zdawałoby się różnice w sformułowaniu definicji, powołanie się na inne źródła lub akty prawne w dokumentowaniu definicji, mogą pociągnąć za sobą istotne różnice zarówno w interpretacji danych, jak i w samych danych. Centralna koordynacja tych prac powinna być więc respektowana przez wszystkich zainteresowanych w ich własnym interesie.

W zakresie klasyfikacji i nomenklatur funkcje koordynacyjne spoczywają na Głównym Urzędzie Statystycznym, współdziałającym z Komisją Planowania, Ministerstwem Finansów i innymi resortami. W związku z rozwojem resortowych systemów informatycznych coraz pilniejsza staje się potrzeba przejścia na komputerową technikę prowadzenia tego obszaru bazy normatywnej, co wiąże się z koniecznością wprowadzenia pewnych modyfikacji w dziedzinie kodowania oraz organizacji służby konserwacji bazy normatywnej przy wykorzystaniu techniki komputerowej. Istotną barierą stanowią tu środki techniczne, jakimi dysponuje obecnie GUS. Obecny sprzęt komputerowy pozwoliłby jedynie na organizację obsługi w trybie przetwarzania partiiowego z dość długim czasem oczekiwania, co obniża atrakcyjność takiej obsługi dla użytkowników. Do czasu wyposażenia GUS w odpowiedni sprzęt pozostaje więc tradycyjna forma publikacji.

Równocześnie jednak konieczne jest ściśle stosowanie klasyfikacji i nomenklatur SPIS przez systemy obiektowe i resortowe oraz identyfikowanie informacji zgodnie z tymi zasadami. Aktualnie obserwuje się szereg niespójności w tworzeniu pochodnych klasyfikacji i nomenklatur, które są następnie wykorzystywane do identyfikowania informacji pierwotnych w systemach resortowych. Takie działanie przekreśla spójność i porównywalność danych między systemami centralnymi już u samego źródła — na wyjściu z systemu obiektowego. Gdyby pochodne

PRZEGLĄD METOD AUTOMATYCZNEJ POPRAWY BŁĘDÓW

Zazwyczaj błędy w programach dzieli się na:

— leksykalne, polegające na przeliterowaniu w poszczególnych słowach programu, określonych regulach leksyki (ortograficzne, *spelling*); źródłem ich jest niedopatrzenie programisty, pomyłka osoby kodującej program na nośniku maszynowym lub przekłamanie urządzenia do przygotowania danych

— syntaktyczne, wynikające z nieznamomości lub złego stosowania reguł syntaktycznych języka programowania

— semantyczne, obejmujące pozostałe błędy, przy czym często w tej klasie wyróżnia się błędy logiczne powstałe na skutek nieodpowiedniej algorytmizacji rozwiązywanego problemu.

Znane algorytmy automatycznej poprawy błędów dotyczą co najwyżej dwu pierwszych z wymienionych wyżej kategorii.

Podstawą automatycznej poprawy błędów jest istniejący w programie pewien nadmiar informacji, nie wykorzystywany w poszczególnych fazach translacji. Nie udało się jak dotychczas opisać formalnie tego nadmiaru i stąd metody jego wykorzystania mają charakter heurystyczny.

Pierwsze algorytmy automatycznej poprawy błędów [1, 2, 7, 12] nie wykorzystywały kontekstu wykrytego błędu, a co za tym idzie niewiele korzystały z redundancji w programie. Poprawki miały charakter lokalny i w kontekście całego programu mogły okazać się błędne. Poprawa dotyczyła jedynie błędów leksykalnych, a jej podstawą było zwykle przygotowane wcześniej opracowanie statystyczne popełnianych błędów.

Stosowana później najszerzej metoda słownika wzorców [5, 8, 10, 14] wykorzystywała do poprawy błędów pewien niewielki kontekst programu. Kategoria błędów była jednak nadal znacznie ograniczona i dotyczyła przeliterowań w słowach wyróżnionych programem umieszczonych uprzednio w słowniku wzorców.

Znana z ostatnich prac metoda globalnego kontekstu [11] wykorzystuje pełny kontekst programu i uniwersalny analizator syntaktyczny. Tak ogólne rozpatrywanie problemu powoduje konieczność równoległego prowadzenia wielu rozbiórów programu. Wybór poprawek następuje po zakończeniu rozbiórów i zliczeniu liczby wprowadzonych poprawek w każdym z rozbiórów. Równoległość ta prowadzi niestety do nieefektywności metody.

Ten krótki przegląd przekonuje, że dla skonstruowania skutecznej i efektywnej metody automatycznej poprawy błędów należy wykorzystać algorytmy analizy syntaktycznej i generacji poprawek, sterowane składnią języka programowania, natomiast wybór poprawki uzależnić od ograniczonego kontekstu błędu. Zgodnie z tak określonymi założeniami w Środowiskowym Ośrodku Informatyki Politechniki Poznańskiej został skonstruowany algorytm automatycznej poprawy błędów [9], który wykorzystuje ograniczony, lecz elastycznie traktowany kontekst błędu. Do wykrywania błędów został wykorzystany uniwersalny algorytm analizy syntaktycznej J. Earleya [3]. Wybór poprawki dokonywany jest na podstawie jej podobieństwa z analizowanym napisem [9]. Algorytm poprawy cechuje uniwersalność dla wszystkich języków bezkontekstowych. Dokonuje on poprawy błędów leksykalnych i syntaktycznych. Skuteczność poprawy dla tych kategorii błędów oraz uniwersalność algorytmu stanowią, że jest on pewnym „novum” wśród znanych dotychczas algorytmów poprawy.

SYSTEM SAPBS

Danymi wejściowymi dla Systemu Automatycznej Poprawy Błędów Składniowych (SAPBS), realizującego algorytm automatycznej poprawy błędów z wykorzystaniem ograniczonego kontekstu, są analizowany napis (program) i opis gramatyki języka programowania. SAPBS wykorzystuje następujące struktury danych:

- 1) opisujące gramatykę języka, na których dokonuje wyłącznie operacji odczytu
- 2) tworzone przez SAPBS, na których dokonuje operacji zapisu i odczytu, zgodnie z algorytmami analizy i poprawy błędów.

Działania na tych strukturach pozwoliły na zrealizowanie SAPBS w postaci programu napisanego w języku FORTRAN dla maszyn serii ODRA 1300 o nazwie USPB (Uniwersalny System Poprawy Błędów).

Program ten zawiera:

- opis gramatyki języka w segmencie BLOCK DATA
- SAPBS w postaci segmentu głównego.

Segment główny składa się z następujących trzech bloków funkcjonalnych:

- Przygotowanie Warunków Początkowych (PWP),
- Analiza Syntaktyczna (AS),
- Poprawa Błędów (PB).

Bloki te realizują kolejno następujące procedury:

1) PWP — wprowadzenie i zakodowanie analizowanego napisu oraz przygotowanie początkowych wartości struktur danych tworzonych przez SAPBS

2) AS — rozbiór gramatyczny analizowanego napisu

3) PB — generację poprawek i wybór najbardziej odpowiedniej.

Wynikami działania programu USPB mogą być komunikaty:

1) PROGRAM POPRAWNY — w przypadku akceptacji napisu przez analizator syntaktyczny

2) PROGRAM ZAWIERA BŁĄD i propozycje poprawy błędu

3) PROGRAM JEST ZA DŁUGI LUB BRAKUJE ZNAKU KONCA, gdy analizowany napis przekracza założoną w realizacji długość.

Propozycje poprawy błędu podawane są w postaci następującego ciągu znaków:

000C₁000C₂000C₃000C₄,

gdzie C_i (i = 1,2,3,4) są kolejnymi znakami poprawki. Wprowadzana jest także informacja o miejscu błędu i wartości podobieństwa [9] poprawki. Gdy żadna z poprawek nie spełnia kryterium minimalnego podobieństwa [9], poprawki poprzedzane są komunikatem:

NIE ZOSTAŁA ZNALEZIONA AUTOMATYCZNIE POPRAWKA

Dla ilustracji metody oraz sprawdzenia skuteczności zaproponowanego algorytmu został przygotowany język zapytań, przeznaczony dla systemu gromadzenia i przetwarzania danych w rozwiązywaniu problemów kardiologii (opracowany w ramach problemu węzłowego 10.4). Składnię języka pytań opisuje gramatyka bezkontekstowa (tabl. 1).

Tabela 1. Zbiór produkcji P gramatyki języka pytań

```
<ZADANIE> ::= <PYTANIE>;
<PYTANIE> ::= COUNT : <WARUNEK>
<PYTANIE> ::= PRINT : <WARUNEK>
<WARUNEK> ::= <WAR1>
<WARUNEK> ::= <WAR2>
<WARUNEK> ::= <WAR3>
<WARUNEK> ::= <WAR4>
<WAR1> ::= (<NAZWA POLA> <ZR> <ZAWARTOSC>)
<WAR2> ::= (<WAR21>)
<WAR21> ::= <WAR1> OR <WAR 1>
<WAR21> ::= <WAR1> OR <WAR21>
<WAR3> ::= (<WAR31>)
<WAR31> ::= <WAR2> AND <WAR2>
<WAR31> ::= <WAR2> AND <WAR31>
<WAR4> ::= (<WAR41>)
<WAR41> ::= <WAR3> OR <WAR3>
<WAR41> ::= <WAR3> OR <WAR41>
<NAZWA POLA> ::= PROBLEM
<NAZWA POLA> ::= BADANIA
<NAZWA POLA> ::= EKG
<NAZWA POLA> ::= RTG
<NAZWA POLA> ::= DZIAŁANIE
<NAZWA POLA> ::= OCENA
<NAZWA POLA> ::= LABORATORIUM
<NAZWA POLA> ::= WYWIAD
<ZR> ::= =
<ZR> ::= <
<ZR> ::= >
<ZAWARTOSC> ::= <SYMB>
<ZAWARTOSC> ::= <SYMB> <ZAWARTOSC>
<SYMB> ::= #
```


Tabela 2. Przykładowe wyniki programu USPB

PROGRAM:	
1	P
2	R
3	I
4	N
5	T
6	:
7	<
8	R
9	T
10	G
11	>
12	0
13)
14	:

PROGRAM POPRAWNY 939	
PROGRAM:	
1	C
2	O
3	U
4	N
5	T
6	:
7	P
8	R
9	O
10	B
11	L
12	E
13	M
14	<
15	C
16	O
17	S
18)
19	:

PROGRAM ZAWIERA BŁĄD.

BŁĄD ZOSTAŁ ZNALEZIONY PO ANALIZIE 7 SYMBOLI

PROPONOWANA POPRAWKA : 000C000P000R0000

MA PODOBIENSTWO RÓWNE 3

WINNĄ BYĆ WSTAWIONA PO 6 SYMBOLU PROGRAMU.

Przykładowe zdania tego języka oraz napisy błędne zostały wprowadzone jako dane dla programu USPB. Postaci wyników w przypadku akceptacji zdania języka oraz generowania poprawek dla wykrytych błędów przedstawia tabl. 2. Zestawienie uzyskanych w trakcie badań rezultatów ujmuję tabl. 3.

Prezentowany powyżej system SAPBS był próbą objęcia możliwie szerokiej klasy błędów w uniwersalnym systemie, dającym się w prosty sposób wykorzystać dla różnych języków bezkontekstowych o niewielkim zbiorze produkcji gramatyki.

Podjęcie tematyki automatyzacji poprawy błędów programowania, ważnej nie tylko dla początkujących programistów i niedoświadczonych użytkowników, ale również, w kontekście efektywności wykorzystania maszyn cyfrowych, dla wszystkich ośrodków obliczeniowych, wydaje się bardzo istotne zarówno pod względem teoretycznym — znajduje się bowiem w obszarze badań nad automatyczną generacją programów — jak i zastosowań praktycznych może bowiem pozwolić na tworzenie nowoczesnych systemów programowania absorbujących w mniejszym stopniu czas programisty i systemu komputerowego w trakcie uruchamiania programów. Wystarczy tu przytoczyć rezultaty uzyskane w systemie CORC [5], zrealizowanym według kryterium najszybszej poprawy i obejmującym co prawda dosyć wąską klasę błędów, w którym po zastosowaniu automatycznej poprawy błędów średnia liczba próbnych uruchomień programów, jak podają autorzy, zmalała z około kilkunastu do 1,1.

Oprócz korzyści mierzonych oszczędnością czasu komputera i programisty, jaką dają systemy automatycznej po-

Tabela 3. Zestawienie analizowanych napisów i wyników działania systemu SAPBS

Analizowany napis	Poprawność napisu	Teksty poprawek	P	Miejsce
PRINT: (RTG>0);	PROGRAM POPRAWNY			
COUNT: (((PROBLEM = SERCE) OR (PROBLEM = ZAWAL)) AND ((WYWIAD<0) OR (EKG>1)));	PROGRAM POPRAWNY			
RINT: (RTG=1);	PROGRAM ZAWIERA BŁĄD	PRIN	3	0
PRINT: (RTG<1);	PROGRAM ZAWIERA BŁĄD	INT:	3	2
PRINT: (OCENA = 123456);	PROGRAM ZAWIERA BŁĄD	RINT	4	1
PRIN: (PROBLEM = AS);	PROBLEM ZAWIERA BŁĄD	T: (P	3	4
COUNT: PROBLEM <COS);	PROGRAM ZAWIERA BŁĄD	(PRO)	3	6
COUNT: (ZADANIE>3);	PROGRAM ZAWIERA BŁĄD	BADA	3	7
PRINT: (ASG=0);	PROGRAM ZAWIERA BŁĄD	Patrz tabul.		

prawy błędów, mogą one także znaleźć interesujące zastosowanie w nauczaniu programowania oraz stanowić narzędzie ułatwiające stosowanie komputerów przez niedoświadczonych użytkowników.

LITERATURA:

- [1] Blair Ch. R.: A program for correcting spelling errors. Inf. and. Contr., No 3, 1960, str. 60—67
- [2] Damerou F.: A technique for computer detection and correction of spelling errors. CACM, vol. 7, No 3, March 1964, str. 171—175
- [3] Earley J.: An efficient context — free parsing algorithm. CACM, vol. 13, No 2, 1970, str. 94—102
- [4] Feyock S., Lazarus P.: Correction of syntax errors in computer programs. Dept. Information and Computing Sciences, Univ. Oklahoma (opracowanie wewnętrzne)
- [5] Freeman D. N.: Error correction in CORC. AFPIS Conf. Proc. vol. 26, 1997
- [6] Gries D.: Compiler construction for digital computers. John Wiley Sons, Inc., New York, 1971
- [7] Irons E. T.: An error-correcting parse algorithm. CACM, vol. 6, No 11, Nov. 1963, str. 669—673
- [8] Klein M., Levy J. P., Limousin P.: Projet SCARABEE. Centre d'Enseignement Supérieur des Affaires de Jouy-en-Josas. Cahier de recherche, No 10, Jan. 1974
- [9] Koperski J.: Automatyczna poprawa błędów składniowych w programach sterowana gramatyką języka programowania (rozprawa doktorska). Środowiskowy Ośrodek Informatyki, Politechnika Poznańska, 1978
- [10] Levy J. P.: Automatic handling of syntax-errors in SCARABEE: An interactive system. Centre d'Enseignement Supérieur des Affaires de Jouy-en-Josas. Cahier de Recherche, No 2, 1973
- [11] Lyon G.: Syntax-directed least-errors analysis for context-free languages: A practical approach. CACM, vol. 17, No 1, Jan. 1974, str. 3—14
- [12] Morgan H. L.: Spelling correction in systems programs. CACM, vol. 13, No 2, Febr. 1970, str. 90—94
- [13] Nagy G., Pennebaker M. C.: A step toward automatic analysis of student programming errors in a batch environment. Int. J. Man-Mach. Stud., vol. 6, No 5, 1974, str. 563—578
- [14] Subieta K.: Korekcja pojedynczych błędów w wyrazach na podstawie słownika wzorców. INFORMATYKA, nr 2, 1976, str. 15—18.
- [15] Youngs E. A.: Human errors in programming. Int. J. Man-Mach. Stud., vol. 6, No 3, 1974, str. 361—376

Czynniki wpływające na metodykę projektowania systemów informatycznych

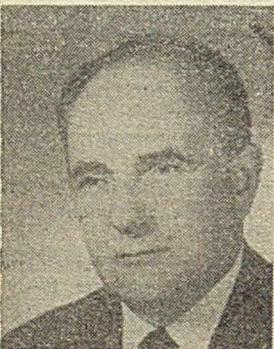
Każde zastosowanie komputerów w gospodarce narodowej, bez względu na to czy dotyczy obliczeń technicznych lub naukowych, czy też służy potrzebom zarządzania, wymaga poprzedzenia fazą projektowania. W fazie tej określane są konstrukcja systemu informatycznego, technologia jego opracowania oraz sposób wdrożenia i użytkowej eksploatacji. Ogół metod postępowania, związanych z opracowaniem systemów i przygotowaniem ich do praktycznego wykorzystania, nazywamy **metodyką projektowania systemów informatycznych**. Metodyka ta różni się w zasadniczy sposób w zależności od rodzaju zastosowań. W przypadku zastosowania komputerów do obliczeń technicznych lub naukowych jest ona stosunkowo prosta i w większości przypadków nie wymaga skomplikowanej dokumentacji projektowej. Precyzuje ona po prostu przedmiot i funkcje zastosowania oraz założenia do opracowania jednego lub kilku programów służących do rozwiązania problemu na określonym komputerze.

Zagadnienie komplikuje się poważnie w przypadku projektowania systemów informatycznych na potrzeby zarządzania. Tryb postępowania jest tu odmienny, gdyż system taki musi być „wmontowany” w cały system zarządzania daną jednostką organizacyjną (lub zgrupowaniem tych jednostek). Dokumentacja projektowa jest znacznie bogatsza, ponieważ określa wszystkie elementy zasilania systemu, wyników przetwarzania oraz sposób ich wykorzystania na potrzeby zarządzania.

Już na wstępie należy stwierdzić, że metodyka projektowania systemów informatycznych na potrzeby zarządzania ma niezmiernie istotny wpływ na rozwój informatyki w kraju. Ośrodki metodologiczne opracowujące metodykę powinny zdawać sobie sprawę z korzyści, jakie przynosi praktyce właściwa metodyka projektowania. Korzyści te można określić następująco:

- znaczne przyspieszenie prac wdrożeniowych
- ujednoczenie trybu postępowania przy projektowaniu i wdrażaniu systemów informatycznych
- uporządkowanie i zwiększenie czytelności dokumentacji projektowej i eksploatacyjnej.

Wszystkie te elementy mają szczególne znaczenie w dobie coraz częstszego przechodzenia od projektowania systemów indywidualnych do systemów uniwersalnych (typowych), przeznaczonych do powszechnego stosowania.



Doc. dr Zdzisław ZAPOLSKI studiował w latach 1947–1951 w Szkole Głównej Planowania i Statystyki na Wydziale Ekonomiki Produkcji. Stopień doktora nauk ekonomicznych uzyskał w roku 1966 w Wyższej Szkole Ekonomicznej w Katowicach. Od wielu lat specjalizuje się w zagadnieniach organizacji i zarządzania przemysłem oraz zastosowania techniki komputerowej w zarządzaniu. Od kilkunastu lat pracuje w Ośrodku Badawczo-Rozwojowym Informatyki. Przez wiele lat był głównym projektantem systemu informatycznego rachunku kosztów produkcji, realizowanego w ramach współpracy międzynarodowej krajów socjalistycznych. Ostatnio pracuje nad Systemem Informatycznym Rachunkowości. Wydał szereg publikacji książkowych i innych.

Należy podkreślić, że metodyka projektowania systemów informatycznych powinna być stale rozwijana i doskonalona. Z jednej strony powinna ona stymulować rozwój informatyki, a z drugiej — formułować nowe reguły postępowania, które zrodziły się w wyniku tego rozwoju i uznane zostały na danym etapie za optymalne.

METODYKA PROJEKTOWANIA SYSTEMÓW INDYWIDUALNYCH

Do niedawna dominowała w kraju metodyka projektowania indywidualnych systemów informatycznych, przeznaczonych dla konkretnych użytkowników. Wynikało to stąd, że ośrodki projektowe wykonywały systemy przede wszystkim dla określonych przedsiębiorstw lub instytucji w dostosowaniu do ich warunków techniczno-organizacyjnych i określonych potrzeb.

Ten kierunek prac projektowych jest typowy dla początkowej fazy rozwoju informatyki oraz braku tzw. systemów firmowych, opracowanych przez producentów komputerów lub przez wyspecjalizowanych producentów oprogramowania. Kierunkowi temu sprzyjał znacznie zróżnicowany park maszynowy, który wymagał specyficznych rozwiązań projektowych dla każdego typu komputera.

Metodyka projektowania systemów indywidualnych, która w dalszym ciągu jest dość szeroko rozpowszechniona, przewiduje następujące główne etapy prac projektowo-wdrożeniowych:

- określenie zadania projektowego
- założenia systemu (projekt wstępny)
- projekt techniczny
- oprogramowanie
- dokumentacja eksploatacyjna
- wdrożenie systemu
- doskonalenie systemu.

W niektórych warunkach etap oprogramowania jest łączony z opracowaniem dokumentacji eksploatacyjnej systemu i wtedy nazywany jest dokumentacją programowo-eksploatacyjną.

W przypadku projektowania systemów o znacznym stopniu skomplikowania dokumentacja eksploatacyjna opracowywana jest dopiero po próbnym wdrożeniu systemu, po którym następuje eksploatacja użytkowa.

Poniżej w najbardziej zwięzły sposób zostaną scharakteryzowane treści i zadania tych etapów.

Każde przedsięwzięcie projektowe rozpoczyna się od spręczenia przedmiotu informatyzacji, czyli od **określenia zadania projektowego**. Jest ono dokonywane na ogół w wyniku wstępnej analizy potrzeb i możliwości informatyzacji w konkretnej jednostce organizacyjnej. Zadanie to może być również narzucone przez przyszłego użytkownika systemu w oparciu o posiadane rozpoznanie własnych potrzeb i możliwości.

Faktyczne prace projektowe rozpoczynają się od opracowania **założeń systemu**, zwanych niekiedy projektem wstępnym. Założenia te zawierają koncepcję rozwiązania problemu, precyzują sposób zasilania systemu (dane wejściowe), informacje uzyskiwane w wyniku przetwarzania oraz sposób ich wykorzystania na potrzeby zarządzania. Określają one również niezbędne środki techniczne oraz przewidywane korzyści eksploatacji systemu i — jeśli to jest możliwe — efektywność systemu. Ponadto założenia zawierają informacje o sposobie przygotowania danych, częstotliwości przetwarzania, kosztach eksploatacji, przyjętej konstrukcji oprogramowania oraz harmonogram dalszych prac projektowo-wdrożeniowych.

Jest to szczególnie ważny etap, ponieważ powstaje w nim koncepcja systemu informatycznego, która po zatwierdzeniu przez użytkownika staje się podstawą do szczegółowego opracowania.

Fazę prac szczegółowych rozpoczyna **projekt techniczny**. Jest on uszczegółowieniem założeń systemu i jako taki określa ostateczne rozwiązania organizacyjne, stanowiąc podstawę do podjęcia prac programowych. Określa on dokładnie dokumenty źródłowe, dane wejściowe, maszynowe nośniki danych, ostateczną formę i treść dokumentów wynikowych, struktury zbiorów na nośnikach magnetycznych oraz inne istotne elementy założeń. Poza tym zawiera stosunkowo szczegółowe założenia do opracowania programów.

Kolejnym etapem prac jest **oprogramowanie** systemu, realizowane na podstawie projektu technicznego. Jest to najbardziej pracochłonny etap prac projektowych, w wyniku którego powstaje zestaw programów przeznaczonych do stosowania na określonym typie komputera. Programy te uruchamiane są na danych próbnych, a następnie na danych rzeczywistych użytkownika.

Dalszym etapem jest opracowanie **dokumentacji eksploatacyjnej**. Dokumentacja ta określa sposób przygotowania maszynowych nośników informacji, sposób i czasokresy przetwarzania na komputerze, niezbędne środki techniczne dla realizacji poszczególnych operacji oraz sposób wykorzystania informacji wynikowych przez użytkownika. Dokumentacja eksploatacyjna jest zbiorem instrukcji, regulujących powyższe sprawy.

Każdy system informatyczny, służący potrzebom zarządzania, wymaga wdrożenia w warunkach codziennej działalności danej jednostki organizacyjnej. Odbывается się to zazwyczaj w ramach oddzielnego etapu zwanego wdrożeniem systemu. Wachlarz prac związanych z tym etapem jest zróżnicowany i zależy od zakresu tematycznego systemu, powiązań z innymi systemami oraz od stopnia przygotowania przedsiębiorstwa do wdrożenia. Efektem końcowym etapu jest użytkowa eksploatacja systemu.

Etap wdrożenia kończy w zasadzie prace projektowo-wdrożeniowe. Ponieważ jednak każdy system informatyczny, podobnie jak każdy system zarządzania, wymaga stalego doskonalenia i dostosowywania do zmieniających się warunków techniczno-organizacyjnych, stąd w cyklu prac projektowo-wdrożeniowych przewiduje się na ogół dodatkowy etap **doskonalenia systemu**. Jest to etap bardzo ważny, chociaż nie wszyscy organizatorzy i projektanci systemów zdają sobie z tego sprawę. Pomijanie tego etapu jest niekiedy wynikiem wygodnictwa zespołów projektowych, którym zależy na szybkim zakończeniu prac i przejściu do nowych zadań. Niewątpliwym wpływem na taki stan rzeczy wywierają istniejące systemy premiowania, które na ogół nie przewidują rekompensaty finansowej za rozwijanie i doskonalenie wdrożonych systemów.

Przedstawiona powyżej metodyka dla systemów indywidualnie projektowanych jest stosowana w naszej praktyce od szeregu lat i w zasadzie zdała egzamin.

METODYKA PROJEKTOWANIA SYSTEMÓW UNIWERSALNYCH

Systemy uniwersalne, obejmujące rozwiązania typowe, przeznaczone do stosowania przez większą grupę użytkowników (np. przez przedsiębiorstwa określonej gałęzi przemysłu) wymagają nieco odmiennej metodyki projektowania. Odmienną ta wynika z innego rodzaju produktu, jakim jest uniwersalny system informatyczny w porównaniu do systemu indywidualnego. Ten odmienny charakter systemu wymaga również odmiennego podejścia do jego budowy.

Przy projektowaniu systemów uniwersalnych na czoło zagadnień wybija się opracowanie odpowiedniego prototypu systemu i wypróbowanie go u kilku użytkowników. W oparciu o próbną eksploatację prototypu opracowywana jest wersja eksploatacyjna systemu, przeznaczona do rozpowszechnienia u użytkowników o założonym profilu działalności.

Specyfika systemów uniwersalnych spowodowała konieczność wprowadzenia dość zasadniczych zmian w stosowanej dotychczas metodyce projektowania systemów informatycznych. Zmodyfikowana metodyka obejmuje następujące etapy:

- opracowanie i uzgodnienie z potencjalnymi użytkownikami koncepcji systemu
- opracowanie i uzgodnienie z potencjalnymi użytkownikami założeń systemu
- opracowanie i wdrożenie w wytypowanych jednostkach organizacyjnych prototypu oprogramowania systemu
- opracowanie wersji eksploatacyjnej systemu
- rozpowszechnienie wersji eksploatacyjnej systemu
- doskonalenie systemu.

Jak wynika z treści powyższych etapów metodyka ta różni się znacznie od metodyki projektowania systemów indywidualnych. Punktem wyjścia jest tu opracowanie **koncepcji systemu** i jej uzgodnienie z przedstawicielami przyszłych użytkowników. Pomija się więc jako wydzielony etap określenie zadania projektowego, co oczywiście nie oznacza, że zadanie takie nie jest precyzowane. Nie wymaga ono jednak podejmowania poważniejszych prac analityczno-badawczych i jest formułowane po prostu w formie hasła wywoławczego, dla którego uzasadnia się podjęcie prac projektowych.

Przezasadnieniem celowości podjęcia prac, koncepcja systemu określa zakres tematyczny, główne kierunki proponowanych rozwiązań, niezbędne środki techniczne, konstrukcję oprogramowania, planowaną pracochłonność i koszty realizacji systemu, przewidywane korzyści oraz organizację i harmonogram prowadzenia prac. Po uzgodnieniu z przedstawicielami użytkowników i zapewnieniu odpowiednich środków realizacji stanowi ona podstawę do podjęcia szczegółowych prac projektowych.

Dalszym etapem jest opracowanie **założeń systemu** lub wytypowanych jego elementów (w przypadku realizacji systemu wielodzielnicowego) i ich ponowne uzgodnienie z przedstawicielami potencjalnych użytkowników. Etap ten realizowany jest na podstawie prac analityczno-badawczych przeprowadzonych w wybranych przedsiębiorstwach. Stopień szczegółowości założeń systemu jest w tym przypadku znacznie większy niż przy systemach indywidualnych, ponieważ stanowią one podstawę opracowania prototypu oprogramowania. Ze względu na pominięcie etapu projektowego technicznego, założenia pod względem szczegółowości powinny odpowiadać projektowi technicznemu, a więc zawierać wszystkie informacje niezbędne do podjęcia prac programowych.

Wykonany w kolejnym etapie **prototyp oprogramowania** jest wdrażany w wytypowanych przedsiębiorstwach. Wdrożenie to ma na celu zbadanie w toku eksploatacji czy przyjęte rozwiązania zaspokajają potrzeby użytkowników. Na podstawie wyników tych badań następuje opracowanie dokumentacji **wersji eksploatacyjnej systemu**, przeznaczonej już do powszechnego stosowania.

Ponieważ system uniwersalny powinien być wdrożony u jak największej liczby użytkowników, metodyka prac projektowo-wdrożeniowych przewiduje oddzielny etap **rozpowszechnienia wersji eksploatacyjnej systemu**. Obejmuje on opracowanie materiałów szkoleniowych, prowadzenie działalności popularyzatorskiej, szkoleniowej i instruktorskiej oraz doradztwo i pomoc użytkownikom przy wdrażaniu systemu.

Również w tym przypadku ostatnim etapem jest **doskonalenie systemu**, które realizowane jest na ogół przez ośrodek projektujący system. Obejmuje ono poza doskonaleniem również konserwację systemu, m.in. dla utrzymania jego zgodności z obowiązującymi aktami normatywnymi. Wyniki tych prac przekazywane są sukcesywnie użytkownikom systemu.

Przykładem zastosowania metodyki projektowania systemów uniwersalnych są podjęte ostatnio w kraju prace nad Systemem Informatycznym Rachunkowości dla przedsiębiorstw przemysłowych. Ze względu na szeroki zakres tematyczny, realizacja tego systemu przewidziana jest na kilka lat.

Należy podkreślić, że omówiona metodyka nie rozwiązuje spraw związanych z zastosowaniem gotowej wersji eksploatacyjnej systemu w konkretnych warunkach działalności użytkownika. Projektowanie wdrożenia wersji eksploatacyjnej u konkretnego użytkownika rozwiązywane jest w oparciu o wytyczne, ujęte w oddzielnym opracowaniu, stanowiącym na ogół część składową dokumentacji tej wersji systemu.

Na projektowanie systemów informatycznych duży wpływ ma konfiguracja dostępnego sprzętu komputerowego, która decyduje o technologii przetwarzania. Dotyczy to zwłaszcza rodzaju urządzeń wejścia/wyjścia oraz rodzaju i pojemności pamięci zewnętrznych. Np. konfiguracje wyposażone w pamięci taśmowe pozwalają realizować tylko systemy charakteryzujące się stosunkowo niedużymi zbiorami informacji, wąskim zakresem tematycznym, bądź przetwarzaniem sekwencyjnym.

Znacznie szersze możliwości realizacyjne stwarzają konfiguracje wyposażone w pamięci dyskowe, zwłaszcza o dużej pojemności. Mogą być na nich uruchamiane systemy o szerszym zakresie tematycznym oraz dużych zbiorach danych, bądź wymagające bezpośredniego dostępu do przechowywanych danych.

Przy projektowaniu systemów uniwersalnych konfiguracja sprzętu komputerowego ma istotny wpływ również na wybór konstrukcji oprogramowania, co w konsekwencji znajduje również odbicie w metodyce projektowania tych systemów.

W praktyce projektowania takich systemów rozróżnić można trzy następujące rodzaje konstrukcji oprogramowania:

- programy parametryczne
- generatory programów
- biblioteka standardowych procedur, oparta na technologii bazy danych.

Systemy oparte na programach parametrycznych mimo znacznych zalet (stosunkowo nieduża pracochłonność opracowania oprogramowania) wychodzą stopniowo z użytku, ponieważ są znacznie mniej ekonomiczne w działaniu od programów opracowanych indywidualnie. Ich przykładem jest angielski system PROMPT¹⁾ lub krajowy PLANTYP, przeznaczony do planowania produkcji złożonych wyrobów.

Systemy oparte na generatorach programów do niedawna stanowiły najbardziej efektywny sposób szybkiego tworzenia oprogramowania użytkowego. Najpowszechniej stosowane systemy firmowe, takie jak PICS²⁾, NIMMS³⁾, MAS⁴⁾, opierają się na generatorach programów, które po

1) PROMPT (Production Reviewing, Organizing and Monitoring of Performance Techniques) — system organizacji i kontroli wykonania produkcji, opracowany przez firmę ICL

2) PICS (Production Information and Control System) — informacyjny system sterowania produkcją, opracowany przez firmę IBM

3) NIMMS (Nineteen Hundred Integrated Modular Management System) — zintegrowany modułowy system zarządzania dla maszyn serii 1900, opracowany przez firmę ICL

4) MAS (Modular Application Systems) — modułowe systemy aplikacyjne, opracowane przez amerykańskiego producenta oprogramowania, firmę Hoskyns Systems Limited

wprowadzeniu parametrów określonego użytkownika i ewentualnych własnych podprogramów pozwalają wygenerować potrzebne programy użytkowe.

Najnowocześniejsza obecnie konstrukcja polega na utworzeniu biblioteki standardowych procedur, opartej na tzw. technologii bazy danych. Instalacja systemu u konkretnego użytkownika polega na wyborze z biblioteki procedur tych, które są potrzebne danemu użytkownikowi oraz na ewentualnym uzupełnieniu ich własnymi podprogramami. W ten sposób w oparciu o rozwiązania standardowe odbywa się stosunkowo łatwo opracowanie indywidualnych systemów dla konkretnych użytkowników. Technologia bazy danych realizowana jest za pomocą uniwersalnego systemu zarządzania bazą danych, który zawiera w sobie języki programowania, umożliwiające użytkownikowi kontaktowanie się z bazą danych.

W ostatnim okresie podjęto również w naszym kraju prace nad realizacją systemów opartych na technologii bazy danych. Prace te prowadzone są w Ośrodku Badawczo-Rozwojowym Informatyki przy zastosowaniu systemu zarządzania bazą danych RODAN.

Jak już wspomniano, wszystkie scharakteryzowane powyżej rodzaje konstrukcji oprogramowania systemów uniwersalnych uzależnione są od konfiguracji sprzętu komputerowego. Najmniejsze wymagania sprzętowe mają programy parametryczne. Konstrukcja oparta na generatorach programów wymaga znacznie większej pojemności pamięci wewnętrznej, a więc zastosowania średnich lub dużych komputerów. Technologia bazy danych może być realizowana tylko na większych komputerach, wyposażonych w system operacyjny typu OS.

Konstrukcja oprogramowania ma zasadniczy wpływ na projektowanie systemów informatycznych. Innego bowiem podejścia do projektowania wymagają programy parametryczne, a innego generatory programów lub technologia bazy danych.

Systemy oparte na programach parametrycznych w podejściu projektowym nie różnią się zbytnio od systemów projektowanych indywidualnie. Realizacja systemów opartych na generatorach programów wymaga odmiennego podejścia do projektowania oraz wdrażania. Stosowane są tu dwie odrębne procedury postępowania: jedna dla zaprojektowania systemu, druga dla wygenerowania programów użytkowych i ich wdrożenia u użytkowników.

Konstrukcja oprogramowania opartego na technologii bazy danych wymaga analogicznego dwuetapowego działania.

Reasumując, należy stwierdzić, że zarówno konfiguracja sprzętu komputerowego, jak i konstrukcja oprogramowania systemu informatycznego, dostosowana do określonego typu sprzętu, mają zasadniczy wpływ na projektowanie systemów na potrzeby zarządzania. Zagadnienia te w większości istniejących materiałów instruktażowo-metodycznych z zakresu projektowania systemów informatycznych są niestety niedostatecznie uwzględnione.

Warunki prenumeraty

Prenumeratę przyjmują oddziały RSW „Prasa-Książka-Ruch” i urzędy pocztowe.

Jednostki gospodarki społecznej, instytucje, organizacje i wszelkiego rodzaju zakłady pracy zamawiają prenumeratę w miejscowych oddziałach RSW „Prasa-Książka-Ruch”, a w miejscowościach, w których nie ma oddziałów — w urzędach pocztowych. Czytelnicy indywidualni opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli.

Cena prenumeraty krajowej wynosi:

- kwartalna — 75 zł
- półroczna — 150 zł
- roczna — 300 zł

Przedpłaty przyjmowane są w następujących terminach:

- do 10 czerwca — na III kwartał i II półrocze
- do 10 września — na IV kwartał

- do 25 listopada — na rok następny, I kwartał, I półrocze
- do 10 marca — na II kwartał

Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW „Prasa-Książka-Ruch”, Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto PKO nr 1531-71 — w terminach obowiązujących dla prenumeraty krajowej.

Prenumerata ze zleceniem za granicę jest droższa od prenumeraty krajowej o 50% dla zlecających indywidualnie i o 100% dla zlecających instytucji i zakładów pracy.

Exemplarze archiwalne czasopism wydawanych przez WCT NOT można nabyć w Dziale Handlowym przy ul. Mazowieckiej 12, 00-048 Warszawa, tel. 26 80 16.

Wyszukiwanie i sortowanie danych za pomocą układu komórkowego

Coraz powszechniejsze zastosowanie metod i środków informatycznych w różnych dziedzinach działalności ludzkiej pozwala na rosnącą precyzję odpowiedzi na zasadnicze dla informatyków pytanie: jaka powinna być charakterystyka systemów informatycznych? Już obecnie można stwierdzić, że znaczna część procesów przetwarzania informacji za pomocą komputerów nie ma charakteru ściśle numerycznego. Ocenia się, że znacząca większość czynności związanych z przetwarzaniem danych przy użyciu maszyn cyfrowych to: sortowanie, wyszukiwanie i uaktualnianie danych. Z tego powodu dużej wagi nabierają wszelkie prace, które stawiają sobie za cel ulepszenie — głównie w sensie skrócenia czasu wykonywania i zmniejszenia kosztów — algorytmów wymienionych działań. Niniejszy artykuł można uważać za próbę w tym kierunku. Został on oparty na pracy [5].

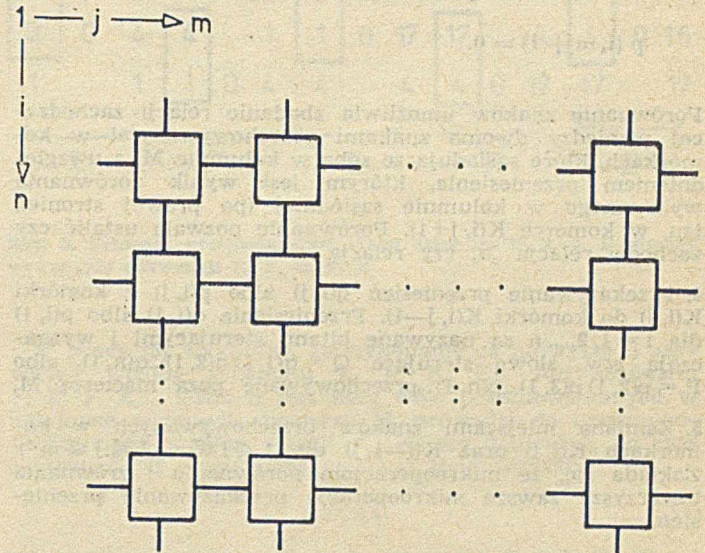
Wyszukiwanie i sortowanie są przedmiotem szczegółowych i wieloaspektowych rozważań, np. [2, 4]. W rozważaniach tych zakłada się, że stosowanym narzędziem jest klasyczna maszyna von Neumanna, a więc taka, która umożliwia jedynie szeregowe wykonywanie obliczenia, a na dodatek miejsca przechowywania danych (pamięć) i ich przetwarzania (arytmometr) są rozłączone. Wydaje się, że możliwości dalszego doskonalenia algorytmów rozważanych działań dla maszyny szeregowej stają się coraz mniejsze. Na przykład dla sortowania metodą porównywania znane jest dolne ograniczenie liczby porównań, która wynosi $kn \lg_2 n$, gdzie n jest liczbą sortowanych elementów, zaś k oznacza pewną stałą nieco większą od jedności [2].

Rozwój technologii układów scalonych skierował zainteresowanie konstruktorów urządzeń cyfrowych m.in. w stronę układów komórkowych (ang. *cellular array*). Podstawowym składnikiem układu komórkowego jest komórka, która zwykle obok zdolności przechowywania danych ma możliwość ich przetwarzania. Komórki są połączone w określony sposób, który wyznacza strukturę układu komórkowego. Układy komórkowe o różnych strukturach stanowiły podstawę do opracowania urządzeń do wykonywania działań o charakterze numerycznym, np. mnożenia i dzielenia [1], oraz działań typu nienumerycznego, np. sortowania [3]. Wydaje się, że zastosowanie układów komórkowych do wykonywania działań nienumerycznych może przybliżyć urzeczywistnienie myśli J. von Neumanna o utworzeniu jednorodnego funkcjonalnie urządzenia, przydatnego do jednoczesnego przechowywania i przetwarzania danych bez konieczności ich przesyłania z pamięci do arytmometru, a przy tym umożliwiającego równoległe wykonywanie wielu działań.

Użycie układu komórkowego o strukturze tablicy prostokątnej umożliwia przechowywanie w jego wierszach słów, które są przedmiotem wyszukiwania i sortowania. Układ ten pozwala na jednoczesne porównywanie wielu par słów. W artykule tym zostanie przedstawiona koncepcja układu komórkowego — nazywanego macierzą M — który służy do wyszukiwania i sortowania słów, przy czym czas wyszukiwania nie zależy od mocy przeszukiwanego zbioru, a czas sortowania jest wprost proporcjonalny do liczności porządkowanego zbioru. Żadne ze znanych urządzeń (i metod) nie odznacza się tak korzystnymi własnościami, zwłaszcza że zarówno podczas wyszukiwania, jak i w czasie sortowania za pomocą macierzy M nie korzysta się z miejsc roboczych pamięci.

STRUKTURA MACIERZY M

Macierz M jest złożona z $n \times m$ identycznych elementów nazywanych komórkami, które są połączone w regularną sieć tworzącą tablicę prostokątną. Tablica ta składa się z n wierszy numerowanych od 1 do n , poczynając od „góry” tablicy, oraz z m kolumn numerowanych od 1 do m , licząc od strony lewej do prawej. Każda komórka jest połączona ze wszystkimi komórkami sąsiadującymi z nią bezpośrednio w wierszu i w kolumnie.



Rys. 1. Struktura macierzy M

Na rysunku 1 przedstawiono schematycznie macierz M . Kwadraty oznaczają komórki, a linie łączące kwadraty symbolizują połączenia występujące pomiędzy komórkami. W celu uproszczenia opisu na rysunku nie zaznaczono rejestrów i sygnałów o stałej wartości dołączonych do komórek „brzegowych” macierzy M .

Komórka, która znajduje się na przecięciu i -tego wiersza dla $1 \leq i \leq n$ oraz j -tej kolumny dla $1 \leq j \leq m$, oznacza się przez $K(i, j)$. Dowolna komórka umożliwia wykonywanie tzw. mikrooperacji. Poniżej zostaną podane mikrooperacje wykonywane w komórce.

Dr inż. Mieczysław MURASZKIEWICZ w 1972 r. ukończył wydział Elektroniki Politechniki Warszawskiej i rozpoczął pracę w Centralnym Ośrodku Informatyki PW. Od 1975 r. pracuje w Instytucie Informatyki Naukowej, Technicznej i Ekonomicznej. Zajmuje się problematyką przetwarzania dużych zbiorów danych, sieciami komputerowymi i strukturami sieci informatycznych.



Mikrooperacje

1. Przechowywanie jednego znaku oznaczanego przez $z(i, j)$. Znakiem może być litera alfabetu łacińskiego, cyfra, itp.¹⁾
2. Zrównanie znaków. Wynik zrównania dla $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$ — oznaczany przez $q(i, j)$ — jest określany według zasady:

$$q(i, j) = \begin{cases} 0, & \text{gdy } b_j \neq z(i, j) \vee q(i, j+1) = 0 \\ 1, & \text{w przeciwnym razie} \end{cases} \quad (1)$$

$$q(i, m+1) = 1$$

gdzie b_j jest znakiem należącym do słowa $x = b_1 b_2 \dots b_m$ znajdującego się poza macierzą M . Zrównanie znaków pozwala ustalić czy dwa znaki są w relacji $=$, czy w relacji \neq przy czym podczas zrównania w komórce $K(i, j)$ uwzględnia się tzw. przeniesienie, którym jest wynik zrównania, wyznaczony w komórce $K(i, j+1)$.

3. Porównanie znaków. Wynik porównania dla $i = 2, 3, \dots, n$; $j = 1, 2, \dots, m$ — oznaczany przez $p(i, j)$ — jest określany według zasady:

$$p(i, j) = \begin{cases} 0, & \text{gdy } z(i-1, j) = z(i, j) \wedge p(i, j+1) = \\ & = 0 \vee z(i-1, j) > z(i, j), \\ 1, & \text{w przeciwnym razie,} \end{cases} \quad (2)$$

$$p(i, m+1) = 0.$$

Porównanie znaków umożliwia zbadanie relacji zachodzącej pomiędzy dwoma znakami przechowywanymi w komórkach, które sąsiadują ze sobą w kolumnie M , z uwzględnieniem przeniesienia, którym jest wynik porównania wykonanego w kolumnie sąsiedniej (po prawej stronie), tzn. w komórce $K(i, j+1)$. Porównanie pozwala ustalić czy zachodzi relacja \geq , czy relacja $<$.

4. Przekazywanie przeniesień $q(i, j)$ albo $p(i, j)$ z komórki $K(i, j)$ do komórki $K(i, j-1)$. Przeniesienia $q(i, 1)$ albo $p(i, 1)$ dla $i = 1, 2, \dots, n$ są nazywane bitami sterującymi i wyznaczają tzw. słowo sterujące $Q = q(1, 1)q(2, 1)\dots q(n, 1)$ albo $P = p(2, 1)p(3, 1)\dots p(n, 1)$ przechowywane poza macierzą M .

5. Zamiana miejscami znaków przechowywanych w komórkach $K(i, j)$ oraz $K(i-1, j)$ dla $1 < i \leq n$, $1 \leq j \leq m$ ²⁾. Zakłada się, że mikrooperacjom porównania i zrównania towarzyszy zawsze mikrooperacja przekazywania przeniesień.

Jak już wspomniano we wstępie, w macierzy M można przechowywać słowa. Słowa są ciągami znaków o długości m , przy czym za bardziej znaczące pozycje słowa uważa się te, które znajdują się po lewej stronie słowa. W dowolnym i -tym, dla $i \in \langle 1, n \rangle$, wierszu macierzy M można przechowywać jedno słowo $z(i, 1)z(i, 2)\dots z(i, m)$ w ten sposób, że w każdej komórce znajduje się jeden znak słowa. Słowo znajdujące się w i -tym wierszu będzie oznaczane przez s_i . Działania wykonywane na słowach będą nazywane operacjami. Dowolna operacja składa się z m mikrooperacji, z których każda odnosi się do jednej komórki.

Rozróżnia się dwa rodzaje operacji: szeregowe i równoległe. Wykonanie operacji szeregowej polega na kolejnym wykonywaniu mikrooperacji w następujących po sobie w wierszu komórkach, poczynając od komórki znajdującej się na najmniej znaczącej pozycji, tzn. kolejno w komórkach $K(i, m)$, $K(i, m-1)$, ..., $K(i, 1)$ dla i oznaczającego numer wiersza. Natomiast wykonanie operacji równoległej polega na jednoczesnym („równoległym”) wykonaniu wszystkich mikrooperacji, które wchodzi w skład tej operacji. Poniżej zostaną podane operacje wykonywane w wierszach macierzy M .

1) Znak będzie dalej nazywany również literą

2) Należy zauważyć, że wśród działań wykonywanych przez komórkę nie ma działań wpisania i wypisania znaku do/z komórki; ograniczenie to przyjęto na użytek niniejszego artykułu w celu uproszczenia opisu

Operacje

• Operacje szeregowe

1. Operacja zrównania słów. Operacja ta składa się z m mikrooperacji zrównania znaków. Biorąc pod uwagę budowę macierzy M oraz znaczenie mikrooperacji zrównania znaków (wzór 1), można pokazać, że wykonanie operacji zrównania słów powoduje wygenerowanie przeniesienia $q(i, 1)$ o wartości 1 w przypadku gdy słowo s_i jest równe zadanemu słowu x znajdującemu się poza macierzą, tzn. $s_i = x$. W przeciwnym razie, tzn. gdy $x \neq s_i$ przeniesienie to ma wartość 0.

2. Operacja porównania słów. Operacja ta składa się z m mikrooperacji porównania znaków. Z budowy macierzy M oraz ze znaczenia mikrooperacji porównania znaków (wzór 2) wynika, że w przypadku gdy $s_{i-1} \geq s_i$, dla $i \in \langle 2, n \rangle$, wykonanie operacji porównania słów powoduje wygenerowanie przeniesienia $p(i, 1)$ o wartości 0. W przeciwnym razie, tzn. gdy $s_{i-1} < s_i$ przeniesienie to przyjmuje wartość 1.

• Operacje równoległe

3. Operacja przechowywania słów. Operacja ta składa się z m mikrooperacji przechowywania znaków. Wykonanie tej operacji nie powoduje żadnych zmian w wierszu: nie zmienia jego stanu³⁾.

4. Operacja wymiany słów. Operacja ta składa się z m mikrooperacji wymiany znaków. Wykonanie operacji wymiany słów powoduje, że słowa s_{i-1} oraz s_i , dla $i \in \langle 2, n \rangle$, zostają wzajemnie zamienione miejscami, tzn. po wykonaniu tej operacji słowo s_{i-1} znajduje się w i -tym wierszu, a słowo s_i — w wierszu $i-1$ -szym.

Określenie działań wykonywanych w macierzy M jest procesem „trzy poziomowym”. Pierwszy poziom wyznacza komórka, z którą związane mikrooperacje. Poziom drugi — to wiersz i operacje. Trzeci poziom wyznacza macierz M . Działania odnoszące się do całej macierzy M są opisywane przez tzw. makrooperacje. Dowolna makrooperacja składa się z n operacji. Wykonanie makrooperacji polega na jednoczesnym („równoległym”) wykonaniu wszystkich wchodzących w jej skład operacji. Poniżej zostaną przedstawione makrooperacje wykonywane w macierzy M .

Makrooperacje

1. Przechowywanie. Makrooperacja ta składa się z n operacji przechowywania słów. Jej wykonanie nie zmienia stanu macierzy M .

2. Zrównanie. Makrooperacja ta składa się z n operacji zrównania słów. Wykonanie zrównania poprzedzone jest określeniem słowa x , które znajduje się poza macierzą M . O macierzy M zakłada się, że jest zbudowana w taki sposób, iż każda litera słowa x jest jednocześnie dostępna we wszystkich komórkach, które znajdują się w kolumnie odpowiadającej co do pozycji tej literze. Wykonanie zrównania polega na jednoczesnym wykonaniu we wszystkich wierszach macierzy M operacji zrównania słów. Wynikiem zrównania jest słowo sterujące $Q = q(1, 1)q(2, 1)\dots q(n, 1)$.

3. Porównanie. Makrooperacja ta składa się z $n-1$ operacji porównania słów. Wykonanie porównania polega na jednoczesnym wykonaniu we wszystkich wierszach macierzy M operacji porównania słów. Wynikiem porównania jest słowo sterujące $P = p(2, 1)p(3, 1)\dots p(n, 1)$.

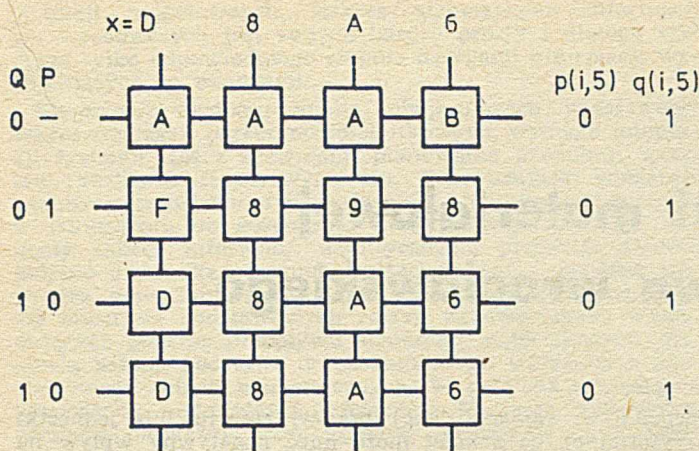
4. Wymiana parzysta. Makrooperacja ta składa się z najwyżej $\left\lfloor \frac{n}{2} \right\rfloor$ operacji wymiany słów. Wykonanie wymiany

parzystej polega na jednoczesnym wykonaniu operacji wymiany słów tylko w tych wierszach o numerach parzystych, dla których $p(i, 1) = 0$; w pozostałych wierszach macierzy M nie wykonuje się żadnych operacji.

5. Wymiana nieparzysta. Makrooperacja ta składa się z najwyżej $\left\lfloor \frac{n+1}{2} \right\rfloor - 1$ operacji wymiany słów. Wykona-

nie wymiany nieparzystej polega na jednoczesnym wykonaniu operacji wymiany słów tylko w tych wierszach o numerach nieparzystych, poczynając od trzeciego wiersza, dla których $p(i, 1) = 0$; w pozostałych wierszach macierzy M nie wykonuje się żadnych operacji.

3) Pojęcie stanu w przybliżeniu odpowiada tutaj zawartości wiersza; zostało ono ściśle zdefiniowane w pracy [5]



Rys. 2. Przykład wykonania makrooperacji zrównania i porównania

Przykład 1. Na rysunku 2 przedstawiono macierz M dla $n = m = 4$. Litery składające się na słowa są cyframi szesnastkowymi (0, 1, 2, ..., A, B, C, D, E, F), a słowa odpowiednio — liczbami przedstawionymi w systemie szesnastkowym. W wierszach przechowywane są kolejno następujące słowa: $s_1 = AAAB$, $s_2 = F898$, $s_3 = D8A6$, $s_4 = D8A6$ — odpowiednie litery słów wpisano na rysunku w kwadraty symbolizujące komórki. Na rys. 2 uwidoczono także znajdujące się poza macierzą M słowo $x = D8A6$ oraz przeniesienia $p(i,5)$ i $q(i,5)$ dla $i = 1, 2, 3, 4$, które są używane odpowiednio przy operacjach porównania i zrównania słów. Łatwo zauważyć, że $x = s_3 = s_4$ oraz $s_1 < s_2$ i $s_2 \geq s_3$. W lewej części rysunku znajdują się słowa sterujące Q oraz P, utworzone po wykonaniu odpowiednio makrooperacji zrównania i porównania. Można zauważyć, że wykonanie makrooperacji wymiany parzystej dla macierzy M, znajdującej się w stanie podanym na rys. 2, spowodowałoby jedynie wzajemną zamianę miejscami słów z trzeciego i czwartego wiersza, zaś wykonanie wymiany nieparzystej dla tego samego stanu doprowadziłoby do wzajemnej zamiany słów z drugiego i trzeciego wiersza macierzy M.

WYSZUKIWANIE I SORTOWANIE

Przyjmuje się, że ciąg słów s_1, s_2, \dots, s_n znajduje się w kolejnych (poczynając od pierwszego) wierszach macierzy M.

Wyszukiwanie

W niniejszym artykule zakłada się, że celem wyszukiwania jest wskazanie tych słów s_k z ciągu przechowywanego w macierzy M, które są równe zadanemu słowu x znajdującemu się poza tą macierzą, tzn. słów spełniających warunek $x = s_k$ dla $k \in \langle 1, n \rangle$. Wyszukiwanie polega na wykonaniu makrooperacji zrównania przy założeniu, że $q(1,5)q(2,5)\dots q(n,5) = 11\dots 1$. Dla wierszy k-tych macierzy M, w których znajdują się słowa spełniające warunek wyszukiwania wartość przeniesienia $p(k,1)$ wynosi 1. Przykład wyszukiwania został podany w poprzednim punkcie (rys. 2). Należy zwrócić uwagę na fakt, że czas wyszukiwania nie zależy od mocy przeszukiwanego zbioru, choć zależy liniowo od długości słów.

Sortowanie

Przyjmuje się, że ciągów słów s_1, s_2, \dots, s_n nazywa się uporządkowanym według relacji $<$, jeśli:

$$(\forall i, j \in \langle 1, n \rangle) (i < j \iff s_i < s_j) \quad (3)$$

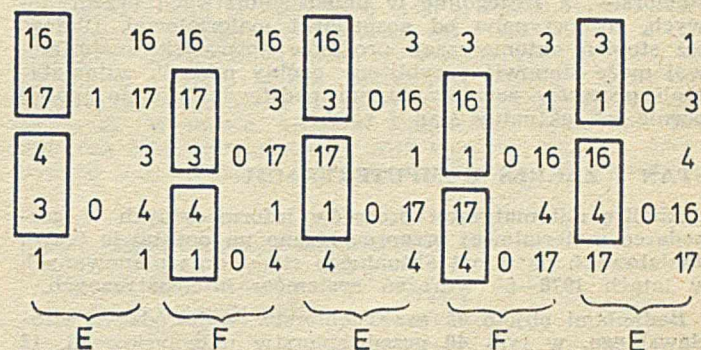
Celem sortowania za pomocą macierzy M jest przekształcenie ciągu słów s_1, s_2, \dots, s_n przechowywanych w tej macierzy w ciąg uporządkowany według relacji $<$. Przedstawienie sposobu sortowania w macierzy M ułatwi określenie tzw. sekwencji. Rozróżnia się dwa rodzaje sekwencji, które będą oznaczone przez E i F.

Wykonanie sekwencji E polega na wykonaniu dwóch makrooperacji: jako pierwszej — makrooperacji porównania i jako drugiej — makrooperacji wymiany parzystej. W przypadku sekwencji F wykonywane są kolejno makrooperacja porównania i makrooperacja wymiany nieparzystej.

Dla obu sekwencji zakłada się, że $p(1,5)p(2,5)\dots p(n,5) = 00\dots 0$. Wynikiem wykonania sekwencji (E lub F) jest zamiana ról poprzedników i następników w tych spośród „utworzonych” w macierzy M par słów, dla których $s_{i-1} \geq s_i$, dla $1 < i \leq n$.

Przykład 2. Rozważany będzie ciąg słów 16, 17, 04, 03, 01⁴⁾ (literami są cyfry dziesiętne, a ich ciągi należy traktować jako liczby dziesiętne), przechowywane w macierzy M dla $n = 5$, $m = 2$.

Na rysunku 3 przedstawiono schematycznie przebieg przekształcania tego ciągu przez kolejne wykonanie sekwencji E, F, E, F, E. Pierwsza kolumna od lewej w każdej sekwencji zawiera słowa przechowywane w kolejnych wierszach macierzy (rozważane pary wpisano w prostokąty). W kolumnie drugiej podano mające wpływ na dalsze działania wartości bitów $p(i,1)$, a w kolumnie trzeciej — ciąg przekształcony przez działanie sekwencji. Z rys. 3 wynika, że ciąg otrzymany jako wynik ostatniego przekształcenia jest ciągiem uporządkowanym według relacji $<$.



Rys. 3. Schemat przekształcania ciągu słów 16, 17, 4, 3, 1 poprzez wykonanie sekwencji E, F, E, F, E

W pracy [5] wykazano, że wykonanie co najwyżej n sekwencji według schematu E, F, E, F, ... lub schematu F, E, F, E, ..., gdzie n jest liczą słów przechowywanych w kolejnych wierszach macierzy M, prowadzi do uporządkowania ciągu według relacji $<$. Należy zauważyć, że czas sortowania T ciągu złożonego z n słów wynosi:

$$T = nt \quad (4)$$

gdzie t jest czasem wykonania sekwencji E (lub F). Oznacza to, że czas sortowania za pomocą macierzy M zależy liniowo od mocy porządkowanego ciągu.

Działanie macierzy M zostało zasymulowane na komputerze IRIS 80. Program symulujący, napisany w języku FORTRAN, oraz wyniki symulacji podano w pracy [5]. Tam też zawarto sugestie realizacji technicznej macierzy M za pomocą układów scalonych.

LITERATURA

- [1] Deverell J.: The design of cellular arrays for arithmetic. The Radio and Electronic Engineer, nr 1, str. 21–26, 1974
- [2] Flores I.: Computer sorting. Mc Grow-Hill, 1969
- [3] Kautz W.: Cellular logic-in — memory arrays. IEEE Transactions on Computers, nr 8, str. 719–727, 1969
- [4] Meadow C.: Analiza systemów informacyjnych. WNT, Warszawa, 1972
- [5] Muraskiewicz M.: Algorytmy procedur nienumerycznych w procesach przetwarzania danych i sposoby ich realizacji w macierzowej jednostce funkcjonalnej. Praca doktorska, IPI PAN, 1978

⁴⁾ Zera występujące na najbardziej znaczącej pozycji będą dalej pomijane

Komputeryzacja gospodarki materiałowej na przykładzie województwa wrocławskiego

Procesy informacyjne realizowane w ramach gospodarki materiałowej (GM) są szczególnie predysponowane do objęcia ich automatyzacją. Przemawiają za tym: masowość danych źródłowych, znaczna ilość stosunkowo prostych decyzji oraz bardzo dużo nieskomplikowanych operacji przetworzeniowych. Wyżej wymienione cechy spowodowały, że automatyzację procesów zarządzania w większości przedsiębiorstw, a szczególnie w przedsiębiorstwach przemysłowych, rozpoczynano od gospodarki materiałowej. Dlatego też stopień automatyzacji procesów gospodarki materiałowej może stanowić przybliżony ogólny miernik automatyzacji procesów zarządzania, na podstawie którego można ocenić ich aktualny stan i rozwój.

STAN I ZAKRES KOMPUTERYZACJI

Analizę automatyzacji procesów informacyjnych w gospodarce materiałowej przeprowadzono na podstawie badań ankietowych na temat aktualnego stanu oraz planowanych w latach 1978—80 wdrożeń systemów informatycznych.

Badaniami objęto 65 przedsiębiorstw województwa wrocławskiego, w tym 40 przedsiębiorstw przemysłowych, 15 budowlanych i 10 transportowych.

Dla celów badawczych problematykę podzielono na 9 podsystemów tworzących następującą strukturę systemu informatycznego gospodarki materiałowej:

- 1) planowanie zużycia
- 2) planowanie zapasów, zaopatrzenia i dostaw
- 3) normowanie zużycia i analiza norm zużycia
- 4) normowanie zapasów
- 5) ewidencja materiałów
- 6) ewidencja przedmiotów nietrwałych
- 7) analiza i sprawozdawczość materiałowa
- 8) kierowanie magazynami
- 9) inne systemy z zakresu gospodarki materiałowej.

Ponad 25% przedsiębiorstw wykazało strukturę systemu gospodarki materiałowej, różną od przyjętej w ankiecie. Odmienność ta sprowadza się do łączenia poszczególnych podsystemów w jeden, wyodrębnienia dodatkowych podsystemów o wyspecjalizowanych funkcjach, jak np. obliczanie normatywnej materiałochłonności wyrobów, zabezpieczenia produkcji w materiały, ewidencjonowanie i rozliczanie kosztów materiałowych, braków, opakowań itp.

Różnorodność struktur wskazuje, że eksploatowane systemy w dużej mierze zostały zaprojektowane pod kątem specyficznych potrzeb indywidualnych użytkowników. Indywidualne rozwiązania w systemach informatycznych mogą być jednak istotną przeszkodą w koordynacji przepływu in-

formacji w ramach danej branży, resortu lub jednostki terytorialnej, co z kolei może mieć negatywny wpływ na integrację procesów sterowania i zarządzania. Negatywny wpływ nieskoordynowanego przepływu informacji można zaobserwować zwłaszcza w dziedzinie gospodarki materiałowej. Utrudniona wymiana informacji pogłębia istniejące bariery materiałowe i powoduje odpowiednie skutki ekonomiczne.

Tabela 1. Eksploatacja systemów GM w przedsiębiorstwach

	Liczba przedsiębiorstw zbadanych	Przedsiębiorstwa eksploatujące systemy GM	
		liczba	%
Przedsiębiorstwa przemysłowe	40	24	60,0
Przedsiębiorstwa budowlane	15	8	53,4
Przedsiębiorstwa transportowe	10	4	40,0
Ogółem	65	36	55,4

Wyniki ankiety wykazały, że tylko 55,4% zbadanych przedsiębiorstw eksploatuje systemy informatyczne gospodarki materiałowej. Jeszcze niższy współczynnik eksploatacji tego typu systemów wykazały przedsiębiorstwa budowlane i transportowe. Niezadowolający jest również współczynnik eksploatacji systemów GM w przedsiębiorstwach przemysłowych.

Tabela 2. Zakres ilościowy eksploatowanych systemów GM w przedsiębiorstwach

Liczba podsystemów GM eksploatowanych w przedsiębiorstwie	Liczba przedsiębiorstw	Odsetek przedsiębiorstw
1—2	23	63,9
3—4	10	27,8
5—6	3	8,3
Ogółem	36	100,0

Bardzo ograniczony jest także zakres ilościowy i tematyczny eksploatowanych systemów GM (tabela 2 i 3).

Przeważająca część przedsiębiorstw (63,9%) z przyjętego w ankiecie układu strukturalnego eksploatuje tylko jeden lub dwa podsystemy GM. Jednocześnie wśród 80 eksploatowanych podsystemów najczęściej eksploatowany jest podsystem ewidencji materiałów, który stanowi około 37,5% ogółu systemów. Znaczny odsetek przedsiębiorstw eksploatuje tylko podsystemy z zakresu ewidencji materiałów i przedmiotów nietrwałych. Oznacza to, że automatyzacją objęto głównie procesy informacyjne zawierające bardzo proste operacje, co oczywiście powoduje mało efektywne wykorzystanie sprzętu komputerowego, a otrzymywane informacje wynikowe są bardzo ubogie.

Po podsystemach ewidencyjnych najczęściej eksploatowane są podsystemy z zakresu analizy i sprawozdawczości materiałowej (15,1%). Rozwój komputeryzacji gospodarki materiałowej w tym kierunku jest w zasadzie poprawny, potwierdza jednak występującą tendencję automatyzacji



Dr inż. Jerzy SZCZUCKI jest absolwentem (1971 r.) Wyższej Szkoły Ekonomicznej we Wrocławiu. Po ukończeniu studiów pracował jako projektant systemów informatycznych w przedsiębiorstwie przemysłowym WZWS „Chemitex”. Od 1973 r. pracuje w Zakładach Naukowo-Badawczych Akademii Ekonomicznej we Wrocławiu na stanowisku kierownika pracowni, a następnie zastępcy dyrektora ds. naukowo-badawczych. Od 1977 r. pracuje jako adiunkt w Instytucie Organizacji i Zarządzania Akademii Ekonomicznej.

operacji najprostszyc. Zakres wymaganych informacji sprawozdawczych jest zwykle bardzo prosty i dlatego wymaga tylko odpowiedniego stopnia agregacji informacji występujących w ewidencji.

Stworzenie systemu analiz, obejmującego podstawowy zakres realizacji przyczynowo-skutkowych, wymaga integracji systemu GM z systemami planowania produkcji, kosztów, rozliczeń finansowych itp. Przeważająca większość badanych przedsiębiorstw, stosując komputeryzację tylko w odniesieniu do gospodarki materiałowej, uzyskuje znikomą zakres informacji analitycznych, przedstawiających jedynie dynamikę strumieni materiałowych i zasobów wewnątrz przedsiębiorstwa. Są to informacje przyczynkowe, nie opisujące podstawowych relacji ekonomicznych występujących podczas przepływu strumieni materiałowych, stanowią jednak podstawę do szczegółowego wyjaśnienia podstawowych relacji ekonomicznych. W procesie zarządzania gospodarką materiałową informacje te mogą być wykorzystane najwyżej przez szczebel operacyjny i to tylko w określonych przypadkach.

Tabela 3. Zakres tematyczny eksploatowanych systemów GM

Rodzaj podsystemu	Liczba podsystemów	Odsetek
1) planowania zużycia	0	11,2
2) planowania zapasów zapotrzebowania dostaw	3	3,7
3) normowania zużycia i analizy norm zużycia	4	5,1
4) normowania zapasów	7	8,7
5) ewidencji materiałów	30	37,5
6) ewidencji przedmiotów nietrwałych	10	12,5
7) analizy i sprawozdawczości	12	15,1
8) kierowania magazynem	—	—
9) inne	5	6,2
Ogółem	80	100,0

Ostatnią, znaczącą grupę podsystemów stanowią podsystemy planowania (ok. 15%) oraz podsystemy normowania (ok. 14%). Ich udział jest jednak zbyt mały, mimo że procedury planowania zużycia zapasów czy zapotrzebowania są stosunkowo łatwe do realizacji na komputerze.

Znamienną cechą jest to, że spośród 65 badanych przedsiębiorstw żadne nie eksploatuje systemów komputerowych kierowania magazynem — i to nawet bardzo duże przedsiębiorstwa prowadzące intensywną gospodarkę magazynową.

Oceniając ogólnie na podstawie omówionej ankiety sytuację w zakresie komputeryzacji gospodarki materiałowej należy stwierdzić, że jest ona niezadowalająca, zarówno pod względem ilości, jak i zakresu tematycznego komputeryzacji tej dziedziny działalności przedsiębiorstw. Nie stwarza to warunków do tak potrzebnego wyzwolenia istniejących rezerw materiałowych.

KOSZTY I PLANOWANE WDROŻENIA

Jednym z problemów szczegółowych, który wyodrębnił w przeprowadzonych badaniach ankietowych, było zagadnienie kosztów pozyskania systemów oraz kosztów eksploatacji.

Jako koszty pozyskania systemu przyjęto nakłady przeznaczone na zakup lub zaprojektowanie systemów, a także ich wdrożenie. Do kosztów eksploatacji zaliczono nakłady związane z wykorzystaniem czasu sprzętu komputerowego, dzierżawę nośników magnetycznych, przygotowaniem maszynowych nośników danych oraz wynagrodzeniem pracowników zatrudnionych przy eksploatacji i konserwacji systemów.

Tabela 4. Koszty pozyskania podsystemów GM

Koszt pozyskania podsystemu (tys. zł)	Liczba podsystemów	Odsetek
5—50	11	13,7
51—100	10	12,5
101—150	9	11,4
151—200	12	15
201—500	11	13,7
501—1000	13	16,3
1001—2000	4	5,0
2001—4000	5	6,3
4001—8000	2	2,5
przekazane nieodpłatnie	3	3,6
Ogółem	80	100,0

Koszty pozyskania systemów wykazują bardzo dużą rozpiętość. Najniższe koszty wynoszą 5 tys. złotych, a najwyższe ponad 7 mln zł. Najwięcej (16,3%) systemów pozyskano ponosząc nakłady w granicach 501—1000 tys. zł, a 13% systemów kosztowało b. mało, bo w granicach 5—50 tys. zł. Jedynie 3 systemy zostały przekazane badanym przedsiębiorstwom nieodpłatnie (uzyskały je przedsiębiorstwa budowlane). Wysokość kosztów pozyskania systemów potwierdza fakt, że większość z nich stanowią systemy opracowane indywidualnie dla poszczególnych przedsiębiorstw.

Tabela 5. Roczne koszty eksploatacji podsystemów GM

Roczne koszty eksploatacji podsystemów w przedsiębiorstwie (tys. zł)	Liczba przedsiębiorstw	Odsetek
do 100	2	5,6
100—500	9	25,0
500—1000	12	33,4
1001—2000	9	25,0
2001—3000	1	2,7
3001—4000	2	5,6
4001—5000	1	2,7
Ogółem	36	100,0

Znikomą część stanowią systemy powielarne, które uzyskano nieodpłatnie lub zaadaptowano stosunkowo niskim kosztem. Faktem wysoce interesującym pozostają jednak bardzo wysokie koszty pozyskania części systemów. Zastanawiające jest wydatkowanie tak dużych nakładów na pozyskanie przeważnie prostych systemów ewidencyjnych lub sprawozdawczych.

Tabela 6. Lokalizacja ośrodków obliczeniowych

Lokalizacja ośrodka	Liczba przedsiębiorstw	Odsetek
Ośrodek przetwarzania zlokalizowany w tej samej miejscowości co przedsiębiorstwo	30	83,4
Ośrodek przetwarzania zlokalizowany w innej miejscowości niż przedsiębiorstwo	6	16,6
Przedsiębiorstwa posiadające ośrodek własny	7	19,4
Przedsiębiorstwa korzystające z ośrodków obcych	29	80,6
Przedsiębiorstwa korzystające z ośrodków sieci ZETO	12	41,4

Rocznie koszty eksploatacji systemów GM wahają się w granicach około 100 tys. zł — 5 mln zł. Największy odsetek przedsiębiorstw (33,4%) ponosi koszty eksploatacji w wysokości 500—1000 tys. zł. Przyjęty przez przedsiębiorstwa tematyczny zakres komputeryzacji w powiązaniu z wysokością kosztów pozyskania oraz eksploatacji podsystemów GM pozwala na stwierdzenie, że w badanych przedsiębiorstwach niska jest efektywność komputeryzacji procesów zarządzania. Realizowana strategia komputeryzacji w dziedzinie gospodarki materiałowej nie pozwala na uzyskanie pełnych efektów ekonomicznych.

Interesujące wyniki uzyskano na podstawie ankiety również w odniesieniu do lokalizacji ośrodków, w których eksploatowano podsystemy GM.

Okazało się, że zaledwie 19,4% przedsiębiorstw ma własne ośrodki, a wśród przedsiębiorstw korzystających z ośrodków obcych ponad połowa korzysta z usług ośrodków sieci ZETO, często znacznie oddalonych od siedziby przedsiębiorstwa (16,6%). Sytuacja taka powoduje, że informacje wynikowe docierają do użytkowników w większości przypadków ze znacznym opóźnieniem i uniemożliwiają zastosowanie systemów do operatywnego sterowania kontroli i analizy przepływów materiałowych.

Zamierzenia badanych przedsiębiorstw w zakresie planowanych wdrożeń obrazuje tabela 7.

Tabela 7. Przewidywane wdrożenia systemów Informatycznych w latach 1978—1980

Rodzaj przedsiębiorstw	Liczba przedsiębiorstw planujących wdrożenia do 1980 r.	Liczba przedsiębiorstw eksploatujących systemy, lecz nie przewidujących nowych wdrożeń	Liczba przedsiębiorstw nie eksploatujących systemów i nie planujących żadnych wdrożeń	Planowany zakres tematyczny wdrożeń		
				podsystemy ewidencyjne	podsystemy planistyczne	inne
Przedsiębiorstwa przemysłowe	16	11	13	12	11	11
Przedsiębiorstwa budowlane	3	5	7	2	—	2
Przedsiębiorstwa transportowe	1	3	6	1	—	—

Ogółem 21 przedsiębiorstw (32%) przewiduje wdrożenie nowych podsystemów do 1980 r. Najwięcej wdrożeń planują przedsiębiorstwa przemysłowe. Wśród przedsiębiorstw eksploatujących obecnie podsystemy 19 (29%) nie przewiduje nowych wdrożeń, a wśród przedsiębiorstw nie stosujących dotąd techniki komputerowej aż 48,3% nie zamierza wdrożyć żadnego podsystemu do roku 1980. Plany wdrożeń przewidują w większości przypadków wdrażania podsystemów ewidencyjnych oraz systemów planistycznych.

Tabela 8. Przewidywane roczne zapotrzebowanie godzin pracy komputera w 1980 roku

Liczba godzin	Liczba przedsiębiorstw	Odsetek
0—50	2	5,2
50—100	5	12,8
101—200	8	20,5
201—300	5	12,8
301—400	8	20,5
401—600	3	7,7
800—1200	3	7,7
1200—1500	5	12,8
Ogółem	39	100,0

Interesujące dane uzyskano również na temat przewidywanego rocznego zapotrzebowania godzin pracy komputera w 1980 roku (tabela 8).

Z danych tych wynika, że około 40% przedsiębiorstw będzie eksploatowało systemy bardzo małe (do 200 godzin rocznie). Systemy duże (ponad 1200 godzin rocznie) będą eksploatowane zaledwie przez 12,8% przedsiębiorstw.

Oceniając zakres komputeryzacji gospodarki materiałowej na podstawie przeprowadzonych badań można wyciągnąć następujące wstępne wnioski:

- niezadowolający jest obecnie zakres wdrożeń podsystemów ewidencyjnych i planistycznych.
- znaczna część wdrożonych podsystemów wymagała dużych nakładów i jest kosztowna w eksploatacji.

Między UNIDATĄ a HONEYWELLEM. Część 1

Początek lat sześćdziesiątych. Francja pod władzą de Gaulle'a chce mieć własną broń termojądrową. Do prac nad nią potrzebne są obliczenia. Francuzi zamierzają kupić w Stanach Zjednoczonych duży komputer firmy CDC. Amerykanie odmawiają.

W taki to bolesny sposób uzmysłowiono sobie konieczność stworzenia niezależnej informatyki francuskiej. Obliczenia co prawda wykonano na maszynie CDC 6600 w Société d'Études et des Mathématiques Appliquées (S.E.M.A.) — ale na przyszłość należało zapobiec podobnym sytuacjom.

W tym czasie trzy firmy zajmują się we Francji komputerami: Compagnie générale électricité (C.G.E.), Compagnie générale de télégraphie sans fil (C.S.F.), najpoważniejsza z nich, Compagnie des Machines Bull (BULL), zatrudniająca w 1963 roku 18 tysięcy pracowników i dysponująca kapitałem wysokości 140 mln franków.

BULL przygotowuje właśnie nowy model komputera GAMMA 60, zdolny zapamiętać 600 tysięcy cyfr w czasie sekundy i grać w brydża. Firma przeżywa jednak poważne kłopoty, spowodowane przez największego konkurenta

na rynku francuskim — IBM. Sprzęt IBM dyskwalifikuje nową serię 300 BULLA. Firma potrzebuje więc pomocy finansowej. Kilka banków europejskich zgłasza gotowość udzielenia pożyczki, ale ówczesny minister finansów, Valéry Giscard d'Estaing, nie wyraża na nią zgody. Torpeduje również rozpoczęte w lipcu 1962 roku rozmowy BULLA z koncernem GENERAL ELECTRIC. Proponuje się za to rozwiązanie francuskie: współdziałanie trzech firm C.G.E., C.S.F. i BULL oraz banku Paribas. Rozmowy prowadzone na przełomie lat 1963 i 1964 kończą się podpisaniem umowy o współpracy, ustanawiającej związki między zainteresowanymi (lutą 1964).

Szybko okazuje się, że sytuacja wciąż nie jest uregulowana. Już w kwietniu kontynuowane są rozmowy z GENERAL ELECTRIC. Przedstawiciele tej firmy docierają do gabinetu premiera. 14 kwietnia 1964 roku porozumienie przedstawia się akcjonariuszom do akceptacji. Ci, nieco zaszokowani sytuacją (spodziewali się układu z C.G.E. i C.S.F.), po pięciogodzinnej debacie akceptują jednak projekt. Układ zostaje ostatecznie podpisany 22 lipca 1964 roku. Przewiduje on wniesienie przez GENERAL ELECTRIC kapitału 210 mln franków i utworzenie trzech towarzystw:

Société Industrielle Bull-General Electric (S.I.B.G.E.) — produkcja sprzętu informatycznego i badania w tej dziedzinie (BULL — 51%, GENERAL ELECTRIC — 49%), Compagnie Bull-General Electric (B.G.E.) — sprzedaż sprzętu (BULL — 40%, GENERAL ELECTRIC — 51%), Société de promotion commerciale Bull (B.G.) — badania rynku i możliwości ekspansji handlowej we Francji i krajach o wpływach francuskich (BULL — 51%, GENERAL ELECTRIC — 49%).

Amerykanie weszli więc do największej komputerowej firmy francuskiej i przejęli kontrolę nad produkcją i sprzedażą sprzętu. Tworząc niezależną informatykę francuską nie można więc już było liczyć na BULLA.

C.I.I. — niezależna informatyka francuska

Po Valerym Giscard d'Estaing nowym ministrem finansów został Michel Debré. W lutym 1966 roku uzyskał on zgodę de Gaulle'a i premiera Pompidou na utworzenie francuskiej informatyki oraz nawiązał niezbędne w tym celu kontakty z przemysłowcami. 18 lipca obradująca pod przewodnictwem de Gaulle'a rada międzyministerialna zaakceptowała plan rozwoju, opracowany przez komisarza ds. planowania François-Xaviera Ortoli i sekretarza generalnego C.O.P.E.P., Pierre'a Audoina. Plan ten, nazwany przez dziennikarzy „Plan Calcul”, stał się punktem wyjścia do dalszego działania.

30 września 1966 roku utworzono Delegaturę ds. Informatyki. Był to organ rządowy, którego kierownik został podporządkowany bezpośrednio premierowi. Delegatura miała rozwinąć narodowy przemysł komputerowy oraz wprowadzić informatykę do administracji państwowej.

W grudniu 1966 r. z inspiracji państwa powstała nowa firma komputerowa — Compagnie Internationale pour l'Informatique (C.I.I.). W jej skład weszły dwie filie dużych grup prywatnego przemysłu: C.A.E., produkująca dotąd na licencji amerykańskiej komputery na potrzeby wojska, nauki i przemysłu: 9010, 9040 i 9080, oraz S.E.A., produkująca komputery francuskie CAB 500, SEA 3900, SEA 4000.

13 kwietnia 1967 roku podpisano pięcioletnią umowę, w której sprecyzowano cele C.I.I. i zasady pomocy państwa. C.I.I. miała wyprodukować nową serię komputerów *oryginalnych i nie mających nic wspólnego z technologią amerykańską*. Państwo miało wypłacić w formie udziału (w ciągu 5 lat) sumę 400 mln franków, udzielić 40 mln franków kredytów na badania (do zwrotu po odniesionym sukcesie) oraz zagwarantowało pożyczki wysokości 125 mln franków. Przemysł prywatny miał podwyższyć kapitał C.I.I. o 66 mln franków. C.I.I. miała wyprodukować cztery modele komputerów, oznaczone symbolami P0, P1, P2 i P3. Pierwszy z nich, P0, średniej wielkości, na obwodach scalonych, przeznaczony na potrzeby zarządzania, miał się ukazać we wrześniu 1968 roku pod nazwą IRIS 50.

Tym, czym w dziedzinie średnich i dużych komputerów miała być C.I.I., w dziedzinie urządzeń peryferyjnych miała zostać firma SPERAC, a w dziedzinie podstawowych elementów elektronicznych — SESCOSEM.

Powstał także (w listopadzie 1966 roku) ważny ośrodek badań i kształcenia kadr informatycznych Institut de Recherche d'Informatique et d'Automatisme (I.R.I.A.), podległy Delegaturze ds. Informatyki. Istniały też inne ośrodki zajmujące się oprogramowaniem, jak np. wspomniana już S.E.M.A.

Odtąd informatyka francuska (wyluczając BULLA i przedstawicielstwa firm zagranicznych, np. IBM) oparta była na schemacie: Delegatura ds. Informatyki — C.I.I. — prywatny przemysł. „Plan Calcul” zaczęło wcielać w życie.

Pierwsze lata C.I.I. nie były łatwe. Należało pogodzić sprzeczne tendencje: C.A.E., skłaniającej się do kontynuowania współpracy z amerykańską firmą SCIENTIFIC DATA SYSTEM i produkcji komputerów na licencji amerykańskiej, oraz S.E.A., zainteresowanej w tworzeniu oryginalnego, własnego sprzętu, co bardziej odpowiadało założeniom C.I.I. W wyniku tych sprzeczności została nieco zmieniona generalna koncepcja rozwoju: model P0 stał się z czasem uproszczonym P1, P2 zniknął w drugim kwartale 1968 roku pojawił się natomiast nowy model 10, 070 według koncepcji amerykańskiej. Aby nie odejść od założonej koncepcji produkcji, konieczne były wielokrotne interwencje Delegatury ds. Informatyki. Dopiero w styczniu 1968 roku ustalono zakresy działania poszczególnych członków C.I.I.

19 marca 1968 r. rząd powziął decyzję o budowie nowoczesnej fabryki sprzętu komputerowego w Tuluzie.

Jednocześnie następowały istotne przesunięcia w przemyśle prywatnym. Kolejne fuzje doprowadziły do wyłonienia się dwóch gigantów: C.G.E. i THOMSONA, który wchłonął m.in. C.S.F. Te dwie firmy będą odtąd odgrywać pierwszoplanową rolę we francuskiej informatyce. Ponieważ w dużej mierze powstały one jako konkurencyjne względem siebie, należało uzgodnić zakresy działania dla każdej z nich. Rozmowy na te tematy, prowadzone (m.in. pod wpływem nacisków Delegatury ds. Informatyki i banku Paribas) od listopada 1968 roku do czerwca 1969, doprowadziły do podziału wpływów (m.in. THOMSON miał zająć się informatyką, C.G.E. — elektrotechniką).

Nowa sytuacja wpłynęła na zmianę struktury C.I.I. Utworzono nową organizację finansową Fininfor. C.G.E. miało w niej 48%, THOMSON — 52%. Przyjęła ona 70% kapitału C.I.I. Pozostałe 30% należało także do przemysłu prywatnego.

W kwietniu 1970 roku do C.I.I. włączono firmę SPERAC, której jedynymi akcjonariuszami stały się C.G.E. i THOMSON. Zmieniło to zamierzoną strukturę przemysłu komputerowego, a ponieważ SESCOSEM, mimo dotacji państwowych, nie zanotowała większych sukcesów (zresztą nie była to jedyna firma zajmująca się podstawowymi elementami elektronicznymi; istniała np. wspólna filia C.G.E. i PHILIPSA — RADIOTECHNIQUE COMPELE), podstawą niezależnej informatyki francuskiej stała się C.I.I.

Mniej więcej w tym czasie uniknięto wejścia amerykańskiej firmy LEASCO do wspomnianego już wcześniej ośrodka tworzenia oprogramowania S.E.M.A., zatrudniającego 2 tysiące osób i mającego 26 mln dolarów obrotu.

S.E.M.A. odczuwała w tym czasie brak realnych wpływów i potrzebę nowych funduszy. W styczniu 1969 roku negocjacje niemalże zostały zakończony sukcesem. Dopiero zdecydowana interwencja Delegatury ds. Informatyki spowodowała rządowe veto. Umożliwiło to zachowanie niezależności w dziedzinie oprogramowania.

W połowie lat siedemdziesiątych francuski przemysł zajmujący się produkcją oprogramowania znajdował się, po USA, na drugim miejscu w świecie.

20 maja 1970 roku ogłoszono, że GENERAL ELECTRIC wycofuje się z informatyki. Mimo wchłonięcia francuskiego BULLA i włoskiej firmy OLIVETTI nie wytrzymał on konkurencji z IBM. Informatykę GENERAL ELECTRIC miała przejąć firma HONEYWELL. Powstała nowa firma — HONEYWELL INFORMATION SYSTEM (H.I.S.), podporządkowana w przeszło 80% HONEYWELLOWI. Do H.I.S. miały więc przejść także akcje BULLA, na co konieczna była zgoda rządu francuskiego. Powstała okazja przejęcia całego francuskiego przemysłu komputerowego i połączenia BULLA z C.I.I. Szereg czynników uniemożliwiło jednak taką operację. Prezydent Pompidou znalazł się w niełatwej sytuacji: właśnie odmówiono przejęcia kontroli nad grupą Jeumont-Schneider przez amerykańską firmę WESTINGHOUSE. Następną odmowa niosła ryzyko amerykańskich przeciwdziałań — i to w momencie, gdy potrzebne były inwestycje zagraniczne.

THOMSON i C.G.E. niechętnie myślą o fuzji i propagują „otwarcie rynku”. C.G.E. prowadzi pertraktacje z GENERAL ELECTRIC w innych dziedzinach i nie chce sobie psuć stosunków z tą firmą. W dodatku kierownictwo BULLA (sytuacja tej firmy nie jest najlepsza, pasywa wynoszą 360 mln franków, a Amerykanie dysponują w niej już 66% kapitału) wyraża zadowolenie z powodu przejścia do HONEYWELLA. Pertraktacje z C.I.I. nie dają rezultatu.

W lipcu 1970 roku rząd francuski wyraża zgodę na przejęcie akcji BULLA przez HONEYWELL. Kapitał dzielony jest w proporcji 50:50, rolę kierowniczą ma HONEYWELL, Francuzi natomiast zachowują prawo pierwokupu w wypadku wyprzedzaży akcji. BULL pozostał więc „amerykański”.

W latach 1967—1970 C.I.I. wyprodukował 4 modele średnich i dużych komputerów dwóch serii: francuskiego IRISA i serii 10000 na licencji amerykańskiej. Odczuwalna stała się potrzeba zwiększenia rodziny komputerów oraz położenia nacisku na rozwój pamięci i zastosowań. Mimo konkurencji osiągnięto założony poziom obrotów. Wyeksportowano komputery do RFN i Włoch oraz sprzedano licencje Rumunii i Węgrom.

Pozostały problemy finansowania własnego rozwoju (wiązało się to ze wspomnianym już wypożyczaniem komputerów oraz konkurencją dużych firm amerykańskich).

Ponieważ pierwszy układ z państwem wygasł w 1971 roku, od stycznia 1970 roku rozpoczęto pertraktacje na temat następnego porozumienia. Zakończono je podpisaniem kolejnej umowy, ważnej do 31 grudnia 1975 roku, która została ostatecznie zatwierdzona 8 stycznia 1972 roku. W jej wyniku utworzono towarzystwo finansowe C.I.L.O.M.I. (Compagnie Internationale pour la Location de Matériel Informatique), o początkowym kapitale 20 mln franków. Państwo zobowiązało się m.in. do wyasygnowania 209 mln franków (w tym 50 mln franków na C.I.L.O.M.I.). Prywatni akcjonariusze mieli zwiększyć kapitał do 100 mln franków (15 mln rocznie na C.I.I. i 5 mln rocznie na SPE-RAC). Poza tym uzgodniono wysokość sumy przeznaczonej na subwencje, kredyty i pożyczki. W celu sprostania wszystkim wymogom konkurencji C.I.I. otrzymała za zadanie nawiązanie współpracy międzynarodowej, przede wszystkim w Europie, w terminie do końca roku 1973.

UNIDATA — kierunek europejski

Możliwość współpracy europejskiej wskazywał już Ortoli w swoim raporcie z lipca 1966 roku. Pierwsze kontakty nawiązano z PHILIPSEM: w 1967 roku w Eidhoven ustalono możliwość porozumienia po wzmocnieniu się C.I.I. W 1968 roku ponownie propozycję współpracy, ale bez widocznych rezultatów.

Innym potencjalnym partnerem była angielska firma International Computers Ltd. (I.C.L.), która miała już na swym koncie sporo sukcesów, m.in. częściowe „wypchnięcie” IBM z W. Brytanii: ok. 50% komputerów na rynku było produktem I.C.L. W listopadzie 1970 roku C.I.I. i I.C.L. powołały w Brukseli wspólne przedsiębiorstwo studiów na temat przyszłej produkcji i jej standaryzacji — Multinational Data. Prowadzone rozmowy nie doprowadziły jednak do dalszych wspólnych akcji. I.C.L., tworząc nową serię komputerów serii P, zaproponowała C.I.I. ich produkcję. Propozycja nie została dobrze przyjęta: nagłe podjęcie produkcji całkowicie obcych, angielskich komputerów nie odpowiadało francuskim zamierzeniom. Odtąd kontakty między obu firmami stopniowo zamierały (ostatecznie zakończyły się w lipcu 1972 roku).

Z kolei zwrócono się w stronę RFN. Związki z NIX-DORFEM były raczej wykluczone: firma podpisała właśnie umowę z AEG TELEFUNKEN i CONTROL DATA. Pozostawał SIEMENS, związany z amerykańską R.C.A. W sierpniu 1971 roku R.C.A. zdecydowała się odejść od informatyki, co zagroziło brakiem licencji w tej dziedzinie dla SIEMENSA. Odczuwano tam również, że produkowany sprzęt jest częściowo przestarzały. SIEMENS był więc „wolny” i szukał partnerów. Jeszcze w sierpniu nawiązano kontakty z C.I.I. Jako formę współpracy zaproponowano nie fuzję, lecz związek uwzględniający niezależność i równość partnerów. Postanowiono razem prowadzić prace badawcze w zakresie sprzętu oraz studia dotyczące dalszych zamierzeń. Nowa wspólna seria komputerów miałyby ukazać się ok. roku 1975. Postanowiono także utworzyć wspólną organizację dla koordynacji sprzedaży produktów obu firm. 1 lutego 1972 roku ukazała się oficjalna informacja o porozumieniu. Wówczas nagle zgłosili się poprzedni rozmówcy: PHILIPS i I.C.L.

Rozmowy z nimi kończą się przyjęciem PHILIPSA do porozumienia, co ogłoszono we wrześniu 1972 roku, w czasie wystawy SICOB. Zawarto umowę wstępną, w której nowa firma o nazwie UNIDATA postawiła sobie jako cel przejęcie (w ciągu pięciu lat) 6% komputerowego rynku światowego i 20% europejskiego. Co do I.C.L., to postanowiono rozważyć jej kandydaturę ok. roku 1980. Do tego czasu I.C.L. miała pozostać, obok firm amerykańskich, jednym z głównych konkurentów w Europie.

Wydawało się, że rozpoczął się nowy rozdział w informatyce zachodnioeuropejskiej. Zarysowała się realna możliwość, popierana m.in. w E.W.G., stworzenia efektywnej konkurencji z dominacją przemysłu amerykańskiego. W czerwcu 1973 roku ogłoszono wyniki dalszych rozmów. UNIDATA miała powstać na zasadzie równości finansowej i dotyczyć wszystkich filii zainteresowanych firm. Powołano by ją w każdym z krajów, każdy z partnerów miałby w niej (w każdym z państw) równą część kapitału. Decyzje w kwestiach finansowych, handlowych i przemysłowych miały być podejmowane jednogłośnie. Sprzęt figurujący w katalogach C.I.I., SIEMENSA i PHILIPSA miał być jak najprędzej wyprzedany. Przewidywano podjęcie prac nad nową serią sześciu (lub dziewięciu) komputerów kompatybilnych wzajemnie oraz z maszynami IBM. Od-

powiedzialność za produkcję wyglądałaby następująco: modele X0 — PHILIPS, X1 i X3 — SIEMENS, X2, X4, X5 — C.I.I. Kwestia urządzeń peryferyjnych miała być wyjaśniona w przyszłości, nie zajmowano się natomiast w ogóle minikomputerami przemysłowymi ani komputerami dla celów wojskowości i telekomunikacji. Integracja finansowa miała być realizowana długofalowo. 4 lipca 1973 roku podpisano tę umowę.

Sprawa UNIDATY we Francji nie była jednak prosta. C.G.E., odgrywająca poważną rolę w C.I.I., poczuła się zagrożona; SIEMENS był jej konkurentem w sprzeczce telekomunikacyjnej i energii jądrowej, PHILIPS — w urządzeniach peryferyjnych i minikomputerach. Minikomputery PHILIPSA potraktowano (również w pewnych kołach rządowych) jako konia trojańskiego, przy pomocy którego Holendrzy chcieli wejść na rynek francuski. Szczególnie zagrożeni poczuli się przedstawiciele średnich i małych przedsiębiorstw produkujących urządzenia peryferyjne.

Zagrożony przez UNIDATĘ HONEYWELL-BULL (H.-B.) wzmógł akcją propagandową wymierzoną przeciw nowej firmie, wykorzystując w tym celu wciąż żywe tendencje proamerykańskie (wzmocnione zmianą na stanowisku ministra przemysłu, którego zakres działania częściowo pokrywał się z zakresem Delegatury ds. Informatyki, a nowy minister — Charbonnel był niezbyt przychylnie nastawiony do UNIDATY).

Sytuację komplikują też spory pomiędzy C.G.E. i THOMSONEM, wywołane pogwałceniem podziału rynku przez tego ostatniego oraz operacjami finansowymi, które spowodowały nawet odwołanie się C.G.E. do Trybunału Handlowego. Spory te przeniesiono na łono C.I.I., co nie odbija się korzystnie na interesach tej firmy.

Minister Charbonnel, biorąc pod uwagę obawy prywatnego przemysłu, zablokował decyzję rządu dotyczącą akceptacji umowy, a tym samym integrację C.I.I. w ramach UNIDATY. Wysunął natomiast inne propozycje, odchodzące od tego rozwiązania. Rząd nie interweniował. Prezydent Pompidou, zawsze popierający sprawę UNIDATY, był w tym czasie poważnie chory.

Czas płynie i tylko Delegatura ds. Informatyki próbuje zwracać uwagę na szkody wynikające z takiej sytuacji. Dopiero 12 lutego 1974 roku zbiera się rada międzyministerialna. Akceptuje ona UNIDATĘ, a zatwierdza ją także Pompidou. Wskazując konieczność dopracowania szczegółów umowy, na dodatkową pomoc w latach 1973 (wstecz) i 1974 przeznacza się 60 mln franków (C.G.E. i THOMSON zgodzili się przeznaczyć 20 mln franków na potrzeby UNIDATY). 26 lutego Charbonnel zatwierdza ostatecznie UNIDATĘ lecz cała sprawa kosztuje go fotel w ministerstwie. Wydaje się, że realizacja UNIDATY posunęła się dalej. Tymczasem, 2 kwietnia 1974 umiera Pompidou, a Francja wchodzi w okres przedwyborczy.

W tym czasie można już mówić o sukcesach C.I.I. Firma ta w ciągu 7 lat zwiększyła pięciokrotnie obroty. W końcu 1974 roku park komputerowy C.I.I. obejmował 2600 cywilnych systemów zainstalowanych lub zamawianych. Udział w rynku światowym wynosił 1%, europejskim — 5,5%, francuskim — 14% (wobec 14,8% H.-B. i 54,5% IBM). Odbywało się to w warunkach ostrej konkurencji, ale przy pomocy państwa. Pomoc ta wraz z subwencjami w latach 1966—1974 wyniosła 1,8 mld franków oraz 2,5 mld w 1975 roku. Wyprodukowano komputery IRIS 45, IRIS 50, IRIS 60 i IRIS 80 (każdy z nich miał swój odpowiednik wojskowy). Dodatkowo wyprodukowano nie przewidziany w „Planie Calcul” minikomputer ogólnego przeznaczenia MITRA 45. Wszystkie te modele pokazano na wystawie SICOB 1971, podczas gdy dwa lata wcześniej przedstawiono jedynie IRISA 50.

W ramach UNIDATY (wspólnie z partnerami) rozpoczęto prace nad komputerami 7.720 (X-0), 7.730 (X-1), 7.740 (X-2) i 7.750 (X-3). Wyjście na rynek 7.720 zapowiedziano na początek 1974 roku, natomiast pozostałe modele miały ukazać się kilka miesięcy później. Montowano także pewne elementy prototypu X-4. W tym samym czasie w produkcji znajdował się komputer C.S.-40, zaprojektowany dla telekomunikacji.

„Plan Calcul” stworzył 5 tys. nowych stanowisk pracy. C.I.I. mieściła się głównie w dwóch regionach — na zachód od Paryża oraz w Tuluzie, gdzie nowo wybudowana fabryka zatrudniała 1700 osób.

Opracował Piotr STRZAŁKOWSKI na podstawie książek J. Jublina i J.-M. Quatrepointa „FRENCH ORDINATEURS — de l'affaire Bull à l'assassinat du Plan Calcul”. Paris 1976

Artykuł niniejszy, opracowany na podstawie publikacji „Programming quality and productivity” T. C. Jonesa (IBM System Journal, Vol. 17, No. 1, 1978), traktuje o metodach pomiaru pracy programisty. Rozważania te, choć pozornie bez jasnych konkluzji, mogą być interesujące dla każdego, kto interesuje się „wydajnością programowania”.

Mierzenie jakości i wydajności programowania.

Część 1

Obserwując rozwój podstawowych dyscyplin wiedzy można powiedzieć, że postęp naukowy jest w dużym stopniu zależny od rozwoju metod precyzyjnego mierzenia. Dlatego właśnie brak precyzyjnych i jednoznacznych jednostek pomiaru jakości i wydajności programowania był (i w gruncie rzeczy jest nadal) przyczyną troski osób odpowiedzialnych za te prace.

W 1972 roku w San Jose Programming Center of IBM Corporation został utworzony zespół, który zajął się badaniami dotyczącymi jakości programowania, wydajności programisty oraz jednostek miary jasno określających te wielkości. Zespół dokonał szczegółowej analizy powszechnie znanych jednostek, używanych do szacowania jakości i wydajności programowania. Otrzymane wyniki pokazały, jak błędne są niekiedy stosowane jednostki i do jakich prowadzi to paradoksów. Dla przykładu jednostka „koszt błędu” dawała najniższe wartości dla programów najbardziej wadliwych, faworyzując nawet przestarzałe technologie, natomiast jednostka „wiersze programu na osobomiesiąc” zaniżała wartość języków wyższego rzędu na korzyść języków typu ASSEMBLER.

W artykule niniejszym zostaną scharakteryzowane niektóre wyniki otrzymane przez wspomnianą grupę badawczą IBM, dotyczące zwłaszcza stosowanych dotąd jednostek pomiaru. Wyniki te potwierdziły również, że nadal istnieją znaczne trudności mierzenia i porównywania jakości i wydajności programowania.

LICZBA WIERSZY PROGRAMU

Podstawowym problemem wszystkich technik pomiarowych związanych z programami komputerowymi jest dokładne zrozumienie określenia „wiersze programu”. Programy bowiem zawierają nie tylko wiersze instrukcji do wykonania, ale też i wiersze komentarzy, deklaracje, a w pewnych przypadkach wiersze rozkazów języka JOB CONTROL czy makroinstrukcji. Konieczne jest więc sprecyzowanie reguł zliczania wierszy. Konwencja stosowana w jednym ośrodku obliczeniowym wymaga dla przykładu zliczenia tylko rozkazów oraz deklaracji, w innym natomiast także komentarzy, wierszy języka JOB CONTROL, czy makroinstrukcji. Nie można powiedzieć, która z metod jest lepsza. Istotne jest tu ujednolicenie reguł zliczenia, w przeciwnym bowiem razie nie ma możliwości porównania wyników.

Kolejny problem pojawia się, gdy mamy do czynienia z programami napisanymi w językach wyższego rzędu. Na przykład w języku PL/1 wierszem może być wszystko, co występuje pomiędzy średnikami, lub to, co jest napisane w pojedynczym wierszu arkusza do kodowania. Tak więc i tu ważne jest ustalenie reguły, według której będzie się odbywało zliczanie.

Aby uniknąć niejednoznaczności z wierszami programu źródłowego, można liczyć bajty lub rozkazy wynikowe. Chociaż metoda ta jest godna polecenia, nie zawsze jest łatwo porównać końcowa wielkość skomplikowanych programów. Dzieje się tak dlatego, że kompilatory różnią się efektywnością, a ponadto indywidualne style programowania mogą oddziaływać wzajemnie na siebie z kompilatorami i nie można zakładać, że w każdym przypadku x wierszy programu źródłowego w danym języku kompiluje się na y wierszy programu wynikowego.

Inne pytanie, to jak traktować zmiany w istniejącym programie. Czy powinny być liczone tylko nowe wiersze, czy wliczać też wiersze bazowe w poprawianym programie. Ze względu na niejednoznaczności w liczeniu wierszy programu, można by zaproponować inne rozwiązanie. Napisane programy określają mianowicie pewne funkcje i liczba wierszy kodujących te funkcje ma mniejsze znaczenie niż koszt realizacji samych funkcji. Dlatego też można rozważać „koszt funkcji”, a nie „koszt wierszy programu”. Dla zbadania takiej wielkości jak „koszt funkcji” konieczne jest jednak ustalenie rodzajów funkcji wykonywanych przez program i sposobu ich zliczania. Obecna metodologia niestety jeszcze nie pozwala na precyzyjne mierzenie tych wielkości.

MIERZENIE JAKOŚCI PROGRAMU

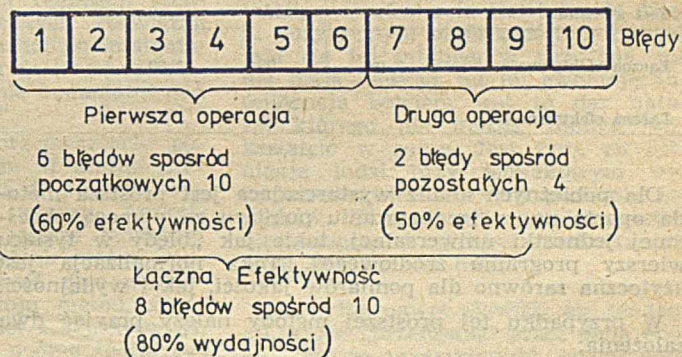
Termin „jakość” należy rozumieć jako brak błędów, które powodowałyby nieprzewidziane zachowanie się programu.

Można wyróżnić dwie podstawowe drogi zmniejszania liczby błędów w programowaniu. Jedną jest zabezpieczenie przed wystąpieniem błędów, drugą — usuwanie błędów, liczby się pojawiły. Stąd mierzenie jakości sprowadza się do pomiarów liczby występujących błędów oraz skuteczności usuwania ich poprzez różne rodzaje testów.

Usuwanie błędów

Podstawowymi wskaźnikami dla pomiaru wydajności przy usuwaniu błędów są „efektywność usuwania błędów” oraz „łączna efektywność usuwania błędów”. Pomiaru te ilustruje rysunek 1.

Ważnym problemem przy usuwaniu błędów jest rozwiązanie złych poprawek, to znaczy takich, które wprowadzają nowe błędy, nie występujące poprzednio w programie. Użyteczne byłoby przechowywanie poprawek, co pozwoliłoby prowadzić oddzielne statystyki dotyczące zarówno wydajności wykrywania i usuwania błędów, jak i skuteczności poprawiania.



Rys. 1. Pomiar łącznej efektywności usuwania błędów

Szacowanie błędów w programie

Ponieważ prowadzenie statystyki wydajności usuwania błędów wymaga poznania wszystkich lub prawie wszystkich błędów znalezionych w programie, wiedza ta może być praktycznym narzędziem przy ocenie programowania. Podczas cyklu rozwojowego programu można przechowywać liczbę błędów znalezionych we wszystkich operacjach testowania, a także liczbę błędów znalezionych w aktualnych jego wdrożeniach. Po zsumowaniu tych wielkości po pewnym czasie znajdujemy zarówno łączną liczbę znalezionych błędów, jak i skuteczność ich usuwania w przeprowadzonych testach. Taka długoterminowa analiza rezultatów wynagradza wysiłki i umożliwia pewien postęp w zakresie efektywności usuwania błędów.

Dodatkowym problemem w mierzeniu wydajności usuwania błędów są zmiany programu. Zmiany wymagań użytkownika dają w wyniku to, że program wchodzący do drugiej operacji testowania może być zupełnie inny niż ten, który wyszedł po pierwszej. Szczegółowe rozpatrywanie tego problemu daje dodatkową analizę źródeł błędów w programowaniu.

Wyróżnia się sześć przyczyn powstawania błędów:

1. trudności opisu funkcjonalnego i niezrozumienie wymagań użytkownika
2. trudności opisu logicznego
3. trudności kodowania
4. trudności dokumentacji
5. błędne poprawki lub wadliwe uzupełnienia
6. inne przyczyny (niewielka grupa).

Przechowywanie danych o procesie projektowania i testowania programów daje ciekawy materiał dla analiz wydajności programowania i efektywności metod testowania. W tabeli 1 podany jest przykład zestawienia tych wielkości dla pewnego hipotetycznego programu.

Tabela 1. Źródła błędów i efektywność ich usuwania

A. Procent błędów wg źródła				
Opis funkcjonalny				15
Opis logiczny				20
Kodowanie				30
Dokumentacja i inne				35
B. Efektywność usuwania błędów (w proc.)				
Czynności	Procent efektywności usuwania błędów w rozbięciu na ich pochodzenie			Procent nieprawidłowych poprawek
	funkcjonalne	logiczne	przy kodowaniu	
Weryfikacja opisu funkcjonalnego	50	—	—	1
Weryfikacja opisu logicznego	40	50	—	2
Sprawdzenie logiczne	60	70	—	2
Sprawdzanie kodowania	65	75	70	3
Test jednostki elementarnej	10	10	25	4
Test funkcjonalny	20	25	25	5
Test modułu	15	20	65	5
Test podsystemu	15	15	55	7
Test systemu	10	10	40	10
Łączna efektywność	98	98	99	
Łączna efektywność		98		

Dla pobieżnych analiz wystarczająca jest prostsza metoda oparta na znormalizowaniu pomiaru za pomocą określonej jednostki uniwersalnej, takie jak „błędy w tysiącu wierszy programu źródłowego”. Taka normalizacja jest użyteczna zarówno dla pomiarów jakości, jak i wydajności.

W przypadku tej prostszej metody należy przyjąć dwa założenia:

— wszystkie błędy, niezależnie od źródła pochodzenia (błędy projektowania, kodowania, czy inne) są zliczane jako pojedyncza wielkość „błędy”

— efektywność usuwania błędów we wszystkich weryfikacjach i testach jest traktowana jako „łączna efektywność usuwania błędów”.

Metoda taka umożliwia zestawienie danych w postaci macierzy (tabeli). Jako jedna oś występuje w niej łączna liczba błędów na tysiąc wierszy programu, a jako druga — łączna efektywność usuwania błędów.

Elementy macierzy można nazwać potencjałem kosztów, przy czym potencjałem kosztów programu jest liczba błędów nie znalezionych podczas operacji testowania. Błędy te są bowiem przyczyną wystąpienia kosztów w czasie dalszego użytkowania programu. Przykładem takiej macierzy o wartościach typowych dla wybranych zakresów zmiennych jest tabela 2.

Tabela 2. Potencjał kosztów jako funkcja łącznej efektywności usuwania błędów i początkowej ogólnej liczby błędów na tysiąc wierszy programu

Ogólna liczba błędów na tysiąc wierszy programu	Procent łącznej efektywności usuwania błędów					
	90	91	92	93	94	95
35	3,5	3,15	2,8	2,45	2,1	1,75
34	3,4	3,06	2,72	2,38	2,04	1,7
33	3,3	2,97	2,64	2,31	1,98	1,65
32	3,2	2,88	2,56	2,24	1,92	1,6
31	3,1	2,79	2,48	2,17	1,86	1,55
30	3,0	2,7	2,4	2,1	1,8	1,5

Widzimy tu, że ogólna liczba błędów w przykładowym programie waha się od 30 do 35 na tysiąc wierszy programu, natomiast łączna efektywność usuwania błędów dla wszystkich weryfikacji i testów waha się od 90 do 95%, co daje zakres potencjału kosztów od 1,5 do 3,5 błędów na tysiąc wierszy programu. Takie wyliczenia można uzupełniać i łączyć z innymi danymi. Dla przykładu program podany w tabeli 2 ma być rozszerzony o 15 000—20 000 wierszy. W najlepszym przypadku dla tego programu otrzymamy 450 dodatkowych błędów (30×15) i dla wydajności 95% potencjału kosztów równy 1,5 błędów na tysiąc wierszy, czyli łącznie 23 błędy. W najgorszym przypadku otrzymujemy 700 błędów oraz potencjał kosztów równy 3,5 błędów na tysiąc wierszy, czyli 70 błędów łącznie.

Postać, w jakiej przedstawia się zakresy zmiennych, nazwano „prostokątem prawdopodobieństwa”, ponieważ ogranicza możliwe zakresy, w których program może być rozwijany.

Tabela 3. Potencjał kosztów jako funkcja wielkości programu

Zakres nie znalezionych błędów na tysiąc wierszy	Wielkość programu (w tysiącach wierszy)	
	15	20
3,5	52,5	70
1,5	18	30

Tabela 3 jest najprostszą formą prostokąta prawdopodobieństwa i pokazuje tylko krańcowe wartości. Trzeba zauważyć, że chociaż żadna ze zmiennych użytych w konstrukcji tego prostokąta nie zmienia się w znacznym stopniu, to różnica między najlepszym (23 potencjalne błędy do znalezienia), a najgorszym wynikiem (70 błędów do znalezienia) jest znaczna.

* * *

Termin „jakości programu” został użyty w znaczeniu: brak błędów. Istnieje oczywiście wiele innych czynników związanych z jakością, ale wskaźniki błędów i efektywności ich usuwania wydają się najlepszymi punktami wyjścia.

Oprac. Ewa JÓZWIAK

Na trzy głosy

Ukazujący się na naszych łamach od prawie roku cykl artykułów poświęconych problemom sztucznej inteligencji wzbudził duże zainteresowanie Czytelników. W numerze 10 z 1978 r. opublikowaliśmy urywki zorganizowanej przez INFOMATYKĘ dyskusji na ten temat. Obecnie przedstawiamy fragmenty najciekawszych wypowiedzi, które zostały nadesłane do redakcji.

Nie ma jednolitej definicji

Nowa gałąź badań, którą przyjęto określać jako „sztuczny intelekt” czy też „sztuczna inteligencja” (SI), zaczęła się rozwijać szczególnie burzliwie w okresie ostatnich dwudziestu lat w oparciu o takie dyscypliny naukowe, jak: matematyka, technika, biologia, psychologia, logika, teoria programowania, teoria algorytmów, teoria sterowania, lingwistyka.

Do dziś jednak trudno jest znaleźć jednolite określenie lub jednolitą definicję sztucznej inteligencji, jak również podać precyzyjne założenia tej nowej, burzliwie rozwijającej się nauki. Najczęściej w gronie naukowców zajmujących się problemami zaliczanymi do sztucznej inteligencji dyskutuje się o zakresie jej zastosowań, tym samym podając intuicyjną, opisową definicję funkcji SI. Twierdzi się nawet, że obecnie nie istnieje jednolita teoria SI, a tym samym nie jest możliwa koordynacja różnych dziedzin badań, jak również nie jest możliwe uzyskanie kompatybilności rezultatów.

Aby dyskutować o problemach wchodzących w zakres zainteresowań SI oraz możliwych metodach podejścia do ich rozwiązywania trzeba odpowiedzieć na pytanie, co to jest inteligencja w ogóle. Można przytoczyć dwie, najczęściej spotykane w słownikach wyrazów obcych czy też encyklopediach, definicje:

- zdolność rozumienia otaczającej sytuacji i znajdowania właściwych, celowych reakcji lub też
- zespół zdolności umysłowych umożliwiających jednostce sprawne korzystanie z nabytej wiedzy przy rozwiązywaniu nowych problemów teoretycznych i praktycznych.

Obie te definicje, aczkolwiek dotyczą tego samego przedmiotu, dają jednak możliwość spojrzenia na problem inteligencji z co najmniej kilku punktów widzenia.

„Zdolność rozumienia otaczającej sytuacji” można utożsamić z procesem percepcji (poznawania) jako funkcji zbierania informacji z otoczenia oraz jej klasyfikacji w celu rozpoznawania i interpretacji. „Znajdowanie właściwych, celowych reakcji” można utożsamić z procesem decyzyjnym. Z kolei „sprawne korzystanie z nabytej wiedzy” wskazuje na aspekty procesu

uczenia się, adaptacji, wyszukiwania informacji. „Rozwiązywanie nowych problemów teoretycznych i praktycznych” wiąże się z heurystycznymi metodami tworzenia algorytmów. Wreszcie „zespół zdolności” wyraźnie wskazuje na powiązania z pojęciem systemu i teorią systemów.

Gdy spojrzymy na dziedziny badań naukowych, to stwierdzimy, że istnieją dziedziny pokrywające powyższy obszar tematyczny:

- teoria automatów abstrakcyjnych, która pozwala na lepsze rozumienie mechanizmów procesu uczenia się, adaptacji i optymalizacji
- teoria grafów, uwzględniająca relacje w postaci związków funkcjonalnych (dzięki badaniom topologicznym staje się możliwe badanie złożonych systemów — np. hierarchicznej struktury centralnego układu nerwowego, itp.)
- teoria gier, określająca optymalne strategie, sposoby postępowania w sytuacjach konfliktowych
- teoria masowej obsługi (czy też teoria kolejek), pozwalająca na ujęcie od strony matematycznej zagadnień związanych z procesem zaopatrzenia przy skończonej liczbie kanałów, jak również optymalizacji tego problemu
- teoria informacji, zajmująca się ba-

daniem procesów przesyłania informacji. Te i inne dyscypliny stanowią teoretyczne lub praktyczne aspekty badań nad inteligencją.

Próbując uściślić zakres tematyczny sztucznej inteligencji można zatem powiedzieć, że: „wszelkie badania dotyczące aspektów związanych (lub spójnych) z problemami inteligencji, prowadzone metodami realizacji technicznej lub teoretycznych rozwiązań wykorzystujących formalizm matematyczny w powyższych dziedzinach wiedzy, nazywa się ogólnie badaniami nad sztuczną inteligencją”.

We wszystkich opracowaniach poświęconych problemom SI można znaleźć pewne nawiązania do powyżej wymienionych dziedzin wiedzy. Aczkolwiek nie mają one jeszcze jednolitej wspólnej teorii, to jednak można, moim zdaniem, wyodrębnić dwa podstawowe kierunki rozwoju metod, środków i teorii większości z aspektów SI. Są to: cybernetyka i ogólna teoria systemów.

Andrzej DZIURNIKOWSKI



Mizerne naśladowictwo

Sztuczna inteligencja — mizerne naśladowictwo mózgu. Nie można jej wymyślić i zaprojektować, ale można dojść do tego drogą kombinacji bionicznej, tak jak to czyniła przez tysiąclecie przyroda.

Jeżeli mówimy o inteligencji, to zakładamy, że przysługuje ona tylko człowiekowi, pamiętajmy więc o innych twórcach przyrody np. delfinach. Aby stworzyć sztuczną inteligencję musimy się zastanowić, co leży u jej podstaw i odpowiedzieć na pytanie czy owa inteligencja to nie instynkt właściwy rozwiniętym twórcom przyrody. Jak odróżnić instynkt od inteligencji? Jak stwierdzić, że nasze działania są skutkiem uczenia społecznego, a nie wynikiem genetycznego, psychofizycznego (dziedzicznego) zaprogramowania?

Spoleczne oddziaływanie, wykształcenie, wywierają niewielki wpływ na inteligencję. Ich rola jest taka, że dzięki pozytywnym okolicznościom pozwalają się tej inteligencji rozwinąć, ale nie mają wpływu na jej ewolucję. Inteligencja bowiem jest to dar natury, którego nie można zdobyć, wykształcić w sobie. Jest tym, co różnicuje ludzi przy jednakowym wykształceniu, jednakowym poziomie zarobków. Objawia się ona dostrzeganiem rzeczy i zjawisk niedostrzeganych dla innych ludzi.

Nie tylko nawet zjawisk, ale tendencji, które „przebijają” przez daną epokę. Jest to przenikliwość, wyobraźnia. Czy można dać automatowi wyobraźnię, albo przenikliwość? Można. Bo

przenikliwość to nie tylko żelazna logika oparta na obserwacjach. To zmysł dedukcji, ale i działania.

Wydaje mi się, że powstaje potrzeba dalszego badania mózgu, a także genów. Bo tylko w oparciu o to można zbudować bionicznego robota, w którym byłaby sprzężona jedność myśli i abstraktu, wyobrażeń i racjonalizmu. Sztucznej inteligencji nie należy utożsamiać z zaprogramowanym robotem, który będzie umiał przeprowadzić inteligentną rozmowę na temat ostatnich wypadków politycznych. Inteligencja to nieuchwytnie oddziaływanie otoczenia, to przyciąganie uwagi. To nie dobrze wyedukowany absolwent wyższej uczelni nawet z medalem, to jest osobowość. Czy uda się stworzyć „sztuczna inteligencję”, która byłaby osobowością? Szczęrze w to wątpię. Inteligencja to nie tylko pamięć. Można mieć ogromną wiedzę i nie myśleć, powtarzać wciąż oklepane banały, stereotypy.

Poziom sztucznej inteligencji będzie zależał od stopnia złożoności jej budowy, od wielkości podstawowych elementów, z których każdy będzie samodzielnym układem programującym. Dopiero wtedy bioniczny układ będzie zdolny do samodzielnego działania twórczego. Właśnie twórczość jest oznaką inteligencji — zdolność do przetwarzania zastanej rzeczywistości. Czy bioniczna inteligencja może tworzyć? Tak i nie: jest wyposażona w lepszą selektywność pamięci, posiada większy zapas słów i skojarzeń, nie posiada jednak wyobraźni i intuicji. A więc czynniki, które są niezbędne w twórczości.

Bioniczna inteligencja będzie racjonalna (wychodząca od przesłanek logicznych), nie irracjonalna. A przecież w każdym racjonalnym podejściu tkwi element irracjonalny, który stanowi załączek racjonalności myśli i postępowania. Inteligencja bioniczna — sztuczna jest możliwa do skonstruowania jeżeli założy się, że będzie: 1) zdolna do syntezy, 2) będzie inteligencją analityczną, 3) będzie inteligencją praktyczną. Dopiero połączenie tych trzech cech daje szansę stworzenia nie kopii ludzkiej inteligencji, ale użytecznego partnera dla człowieka.

Marek GLIŃSKI

Heurystyka pomoże

Proponuję zwrócenie uwagi na bardzo istotną dla badań nad sztuczną inteligencją kwestię heurystyki. Heurystyka (z greckiego: heurisko — znajduję) to reguła postępowania, której celem jest znalezienie rozwiązania określonego zagadnienia, przy czym cechą charakterystyczną każdej heurystyki jest jej zawodność i niepełna określoność. Cechy te wynikają z tego, że heurystyka drastycznie ogranicza poszukiwanie rozwiązań w du-

żych obszarach problemowych. Jednakże to wysoce zredukowane poszukiwanie, czyli rezygnacja z pełnego, systematycznego przeglądania możliwych wariantów, może być uważana za cechę dodatnią heurystyki, jeśli jej wynikiem jest praktycznie zastosowań.

Heurystyka nie gwarantuje optymalnych rozwiązań (choć ich nie wyklucza), co więcej nie gwarantuje ona w ogóle żadnego rozwiązania.

Gwarancję taką zapewnią algorytm, który jednak nie daje rozwiązań idealnych (trzeba uwzględnić koszt i czas). Prosty algorytm otwierania kombinacyjnego sejfu polega na wypróbowywaniu wszystkich kombinacji i sprawdzeniu czy przy każdej z nich sejf się otwiera. Istnieje gwarancja, że jakaś kombinacja otworzy sejf oraz wiadomo, że wszystkie kombinacje wypróbujemy w skończonym przedziale czasu. Jednakże algorytm ten jest bardzo kosztowny i długotrwały na tyle, że chroni sejf nawet przed ludźmi, którzy znają algorytm.

Natomiast przy procesie heurystycznym brak gwarancji, że się rozwiąże zagadnienie, nie jest czymś bezwzględnie złym. Cena braku gwarancji zależy od tego, ile praca kosztuje i jakie istnieją algorytmy alternatywne.

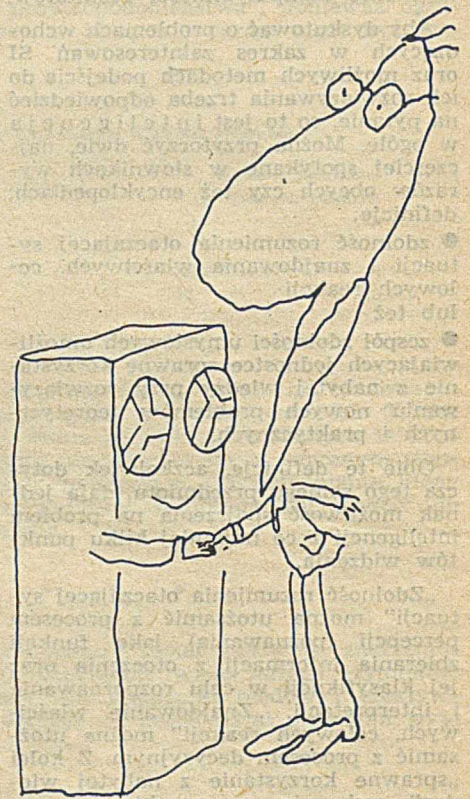
Istnieją problemy, które można i rozwiązuje się algorytmem, np. znajdowanie maksimum różniczkowalnej funkcji — porównuje się tu pierwszą pochodną do zera, a nie bada w sposób heurystyczny każdego punktu na krzywej, bo jest to niemożliwe. Natomiast przy grze w szachy posługujemy się metodami heurystycznymi, ponieważ nikt nie jest w stanie wykonać algorytmu badania wszystkich dalszych ciągów gry do końca. Niemożliwe byłoby wówczas zakończenie nawet jednej partii gry ze względu na czas potrzebny na rozpatrzenie choćby części sytuacji.

Programowanie heurystyczne polega na sukcesywnym podawaniu komputerowi zadań, z których każde następne jest odpowiednio modyfikowane przez człowieka na podstawie zaobserwowanych wyników rozwiązania poprzedniego zadania. W konsekwencji realizowania takiego procesu prób i błędów można dojść do rozwiązania zaspokajającego określone wymogi. Należy tu zwrócić uwagę na pewien pozorny paradoks, programy heurystyczne muszą przypominać algorytmy przystosowane do wykorzystywania w komputerach. Nie opracowano jeszcze języka programowania pozwalającego na posługiwanie się metodami heurystycznymi bez tej „translacji”. Zbieżność ta występuje tylko na poziomie rozpatrywania programów jako zbioru rozkazów dla komputera, nie ma jej już natomiast na poziomie ogólnej strategii stosowanej w programie, która może być strategią algorytmiczną bądź też heurystyczną.

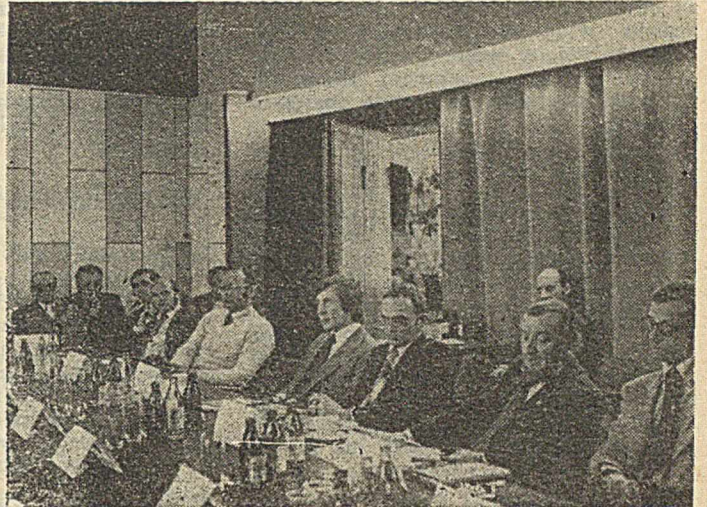
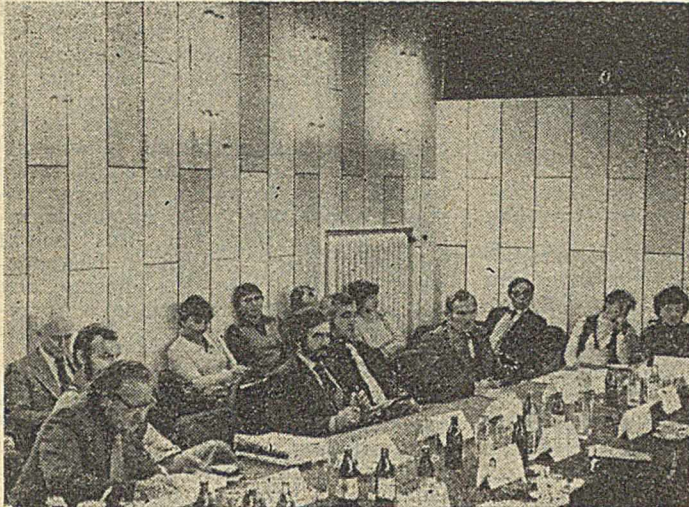
Heurystyka wiąże się z techniką konstrukcji wielkich układów oraz z psychologią pracy. Okazało się, że analiza pewnych współczesnych rodzajów pracy ludzkiej wymaga zastosowania metod heurystycznych.

Zasadę wprowadzenia heurystyki można tu wyjaśnić na przykładzie pracy dyrektora wielkiego kombinatu. Co dzień ma on do czynienia z inną sytuacją i w inny sposób reaguje na proces produkcyjny, nie można więc dokonać pełnego opisu czynności dyrektora w postaci algorytmu. Program heurystyczny musi się przystosować w tym wypadku do zmieniających się bezustannie warunków. Maszyna natrafiając na sytuację, z którą nie miała dawniej do czynienia, musi opracować nową strategię sterowania. Jej działanie musi być oparte na dynamicznym modelu pracy kombinatu, komputer musi „myśleć” kategoriami człowieka. Występuje tu problem, jakie składniki czynności intelektualnej człowieka sterującego układem mogą być na współczesnym etapie rozwoju techniki elektronicznej przyjęte przez maszynę, a jakie nadal musi wykonywać umysł ludzki. Metody heurystyczne mogą bowiem przy obecnym stopniu rozwoju cybernetyki i elektroniki jedynie modelować intelekt ludzki, ale go nie zastępują.

Agnieszka SZEWCZYK



Za czy przeciw systemom powtarzalnym



Spotkanie przy redakcyjnym mikrofonie (listopad 1978) dwóch stron — producentów nowoczesnych systemów informatycznych i ich użytkowników — było rzeczowe, konkretne, chociaż nie brakowało sprzeczności, spór, wzajemnych pretensji. Była to często twarda, męska rozmowa, w której wyjaśniono sobie wiele spraw o istotnym dla informatyki znaczeniu. Wiele zagadnień pozostaje jednak otwartych i wymaga rozwiązania w przyszłości.

Z czterogodzinnej dyskusji drukujemy jedynie niektóre fragmenty wystąpień, dotyczące głównego nurtu omawianych zagadnień. Aby uzyskać logiczną całość nie przestrzegamy tym razem kolejności wypowiedzi, lecz grupujemy je wokół głównych tez omawianych na spotkaniu.

W dyskusji udział wzięli: mgr Janusz Bartczak (Zakłady Metalurgiczne Przemysłu Maszyn Rolniczych „Agromet”), mgr Jerzy Dańda (Instytut Technologii Elektronowej „Unitra-Cemi”), dyr Józefa Hieronimek (Centrala CZSS „Spotem”), mgr inż. Józef Michalak (Fabryka Obrabiarek Precyzyjnych im. 1 Maja „Ponar” — Pruszków), mgr Jerzy Nowak (Kombinat Urządzeń Mechanicznych „Bumar — Łabędy”), mgr inż. Eugeniusz Stencel (MERA-ELWRO), mgr Kazimierz Szumlak (ZETO — Bydgoszcz), mgr Tadeusz Zdziarski (Zjednoczenie Przemysłu Nieorganicznego) i z Ośrodka Badawczo-Rozwojowego Informatyki: dyr mgr Roman Myśliwiec, dr Artur Hajnicz, mgr inż. Jerzy Malicki, mgr Eugeniusz Petryk, mgr Leszek Sankowski.

ZAŁOŻENIE WSTĘPNE: indywidualne systemy informatyczne to już przeszłość...

L. Sankowski: Tezę tę uwydatnia problem nadążania systemu informatycznego za zmianami w systemie zarządzania. Zmiany organizacyjne są nieuniknione. W rezultacie wiele rozwiązań indywidualnych tworzonych przez projektantów systemów i programistów zostało zaprzepaszczone, gdyż życie postępowało szybciej niż dokończyły się zmiany w algorytmach systemu informatycznego. Według badań Rossa ok. 63% błędów w systemach informatycznych ujawnia się w czasie wdrożeń i wstępnej eksploatacji. Wynika z tego, że większość indywidualnych systemów załamuje się już na etapie wdrażania...

A. Hajnicz: Dysponujemy w kraju nadal przeważnie systemami indywidualnymi, lecz ich dalsze produkowanie straciło obecnie sens. Systemy te są nieelastyczne, nieopłacalne, ich produkcja trwa latami.

TEZA 1: W dobrze znanych i opanowanych przez informatykę obszarach lepsze, tańsze, łatwe do adaptacji systemy powtarzalne w coraz większym stopniu wypierają systemy indywidualne.

J. Hieronimek: Systemy powtarzalne uważam za zwrot i nowy kierunek w rozwoju informatyki i zastosowań informatycznych. Wprowadzanie systemów powtarzalnych, ich adaptacja do różnych, konkretnych organizacji, a także ich konserwacja i dalszy rozwój uchroni nas od zbędnego i kosztownego opracowywania wielu systemów w tej samej dziedzinie... Podjęliśmy współpracę z ZETO-Bdygoszcz nad adaptacją ich powtarzalnego systemu finansowo-księgowego. ZETO zaprezentowało swój system w sposób jasny i dzięki temu użytkownicy mogli dostatecznie wcześniej określić, jakie elementy systemu przyjmują bez żadnych zmian, a jakie — szczególnie w dziedzinie zestawień wynikowych — powinny być jeszcze dopracowane...

J. Nowak: Przeszkody w szerokim zastosowaniu systemów powtarzalnych upatruję w tym, że systemy te konstruowane są przez ośrodek ZETO w oparciu o analizę stanu w jednym tylko zakładzie. Opracowuje się system dla jednego użytkownika, a następnie sprzedaje się go innym, często bez uwzględnienia ich specyficznych potrzeb i możliwości. Zachodzi więc konieczność dostosowania się użytkownika do danych i wymogów stawianych przez system powtarzalny, co budzi określone opory...

K. Szumlak: W tej chwili ZETO-Bdygoszcz zarejestrowało ponad 400 wdrożeń systemów powtarzalnych, własnej produkcji. Systemy te są eksploatacyjne w 43 ośrodkach obliczeniowych, w tym — w 19 ośrodkach należących do sieci ZETO. Tylko do września bieżącego (1978) roku dokonaliśmy przeszło stu nowych wdrożeń naszych systemów powtarzalnych...

TEZA 2: Użytkownicy, wyrażając ogólnie zadowolenie z zastosowania systemów powtarzalnych, wysunęli pod adresem producentów i dysponentów tych systemów wiele zarzutów. Dotyczyły one głównie zasad obrotu oprogramowaniem powtarzalnym, a mianowicie:

a) ceny systemów powtarzalnych, zwłaszcza przy ich wielokrotnej sprzedaży kilku przedsiębiorstwom jednego kombinatu

b) udostępniania użytkownikowi przeważnie wyłącznie dokumentacji eksploatacyjnej, a więc sprzedawania — jak mówiono — „czarnej skrzynki”, do której użytkownik nie ma dostępu.

Ceny

J. Nowak: Kupując system powtarzalny dla kombinatu, ponosi się koszty upowszechnienia systemu we wszystkich zakładach kombinatu. Można zatem powiedzieć, że kombinat zmuszany jest do wielokrotnego zakupu tego samego produktu. Wydaje się więc, że należy zmodyfikować zasady obrotu i upowszechniania systemów powtarzalnych w jednostkach gospodarczych, grupujących wiele zakładów lub przedsiębiorstw...

Problem kosztów zakupu i upowszechniania systemów powtarzalnych jest szczególnie drażliwy dla przedsiębiorstw małych, jak też takich organizacji gospodarczych, które skupiają kilka zakładów filialnych. Może należałoby wprowadzić odpłatność proporcjonalnie do liczby zatrudnionych lub przy uwzględnieniu innych mierników? W każdym razie w efekcie kwoty muszą być odpowiednio niższe niż koszt systemu indywidualnego, który by kombinat opracował własnymi siłami...

T. Zdziarski: W wyniku istniejącej polityki cen powstaje niebezpieczeństwo, że producent oprogramowania wdrażając system powtarzalny u 100 użytkowników nie ma czasu i nie jest zainteresowany w rozwijaniu i konserwacji systemu. Opłaty cennikowe preferują działania zmierzające do wielokrotnego wdrażania systemu. Natomiast niskie opłaty za konserwację systemu nie zapewnają użytkownikowi oczekiwanej pomocy podczas eksploatacji i dalszego rozwoju systemu.

J. Hieronimek: Problemem otwartym pozostaje rozwój i konserwacja systemu powtarzalnego. Konserwację powinna prowadzić jednostka autorska. Producent jest najbardziej zainteresowany w rozpowszechnieniu systemu. Konkretny zaś użytkownik jest zainteresowany we wprowadzaniu modyfikacji i uzupełnień, które zaspokoilyby jego nowe potrzeby. Zmieniają się przepisy i zachodzi potrzeba dokonania szybkich zmian w systemie powtarzalnym...

K. Szumlas wyjaśnia stanowisko producentów systemów powtarzalnych w ostatniej kwestii:

Jak rozumiemy pojęcia „konserwacja” i „modyfikacja systemu powtarzalnego”? Pojawia się tu wiele nieporozumień. Pod pojęciem „konserwacja” rozumiemy działalność związaną z usu-

waniem błędów pojawiających się w systemie podczas 12-miesięcznego okresu gwarancyjnego, a także pewne bardzo drobne uzupełnienia. Inną zupełnie sprawą jest modyfikacja systemu powtarzalnego, niezbędna ze względu na zmianę obowiązujących przepisów. Np. ministerstwo wprowadza zmiany do rozliczeń placowych. System placowy wymaga gruntownej modyfikacji, nieraz bardzo pracochłonnej. Przy zmianie przepisów rozliczenia środków trwałych modyfikacja odpowiedniego systemu powtarzalnego angażowała dużą grupę projektantów i programistów w ciągu prawie 1,5 roku. Dajeśmy użytkownikom gruntownie zmieniony, ale sprawdzony, dobry produkt. Kto ma ponosić koszty całego przedsięwzięcia?

„Czarna skrzynka”

J. Michalak: Po zakupie oprogramowania zakład pozostaje w zasadzie sam z wszelkimi problemami wynikającymi z jego wdrażania. Jestem zwolennikiem produkcji i zakupu oprogramowania powtarzalnego, ale takiego, które można samodzielnie rozwijać. Należy też przewidzieć rozbudowę systemu o elementy potrzebne użytkownikowi i przez niego opracowane.

J. Hieronimek: Niejednokrotnie zachodzi potrzeba dokonywania szybkich zmian w systemie i trudno czekać na producenta systemu powtarzalnego. Niezbędne jest zatem dokładne, czytelne udokumentowanie zbiorów i operacji dokonywanych na tych zbiorach. W sytuacji, gdy użytkownik dysponuje pełną, rozszerzoną dokumentacją systemu, może dokonywać zmian w krótkim czasie, co jest niekiedy warunkiem dalszej eksploatacji systemu... Jednostka autorska może się wówczas ograniczyć do nadzoru autorskiego nad zmianami, które użytkownik wprowadza samodzielnie.

J. Nowak: Należy postawić sprawę wyraźnie — kupujemy systemy powtarzalne, ale pod warunkiem, że otrzymamy nie tylko dokumentację eksploatacyjną, ale również dokumentację zbiorów danych, występujących w systemie. Dysponując względnie pełną dokumentacją systemu, użytkownik ma potencjalne możliwości usunięcia błędów programowego po konsultacji telefonicznej z jednostką autorską. Mamy w tym zakresie pewne doświadczenie.

Z tym wiąże się kolejny postulat. Powinniśmy opracować obowiązującą normę dla dokumentacji systemów powtarzalnych — dokumentacji projektowej, programowej, eksploatacyjnej. Mamy już odpowiednie doświadczenie i nie powinno być przeszkód przy opracowaniu polskiej normy określającej kształt i zawartość dokumentacji systemów przeznaczonych do upowszechnienia.

Pełnie odmienny pogląd na te sprawy zaprezentował w imieniu producentów systemów powtarzalnych dyrektor ZETO-Bydgoszcz **K. Szumlas:** W jakiej postaci należy udostępniać użytkownikowi dokumentację systemów powtarzalnych?

Z tych 400 użytkowników naszych systemów powtarzalnych, o których wspomniałem wcześniej, większość nie interesuje się budową zbiorów, obrazem rekordów. Nawet nie wiedzieliby dokładnie co mają dalej z tym fantem robić. Wystarczy im odpowiednio przejrzysta dokumentacja eksploatacyjna. Oczywiście stanowi to pewne ograniczenie dla tych użytkowników, którzy sami chcieliby modyfikować system — dla siebie lub w celu udostępnienia go innym. Zilustruję to na przykładzie. Przyszedł do nas pewien użytkownik i mówi, że już trzy miesiące liczy naszym systemem finansowo-księgowym, jest na ogół zadowolony, ale rażą go nagłówki z firmą innego użytkownika. Okazało się, że ktoś nas wyreczył w obrocie systemem powtarzalnym. Ten nieuczciwy pośrednik łatwy był do rozszyfrowania, ponieważ zostawił swoją wizytówkę na tabulogramach. Za tę usługę ten „ktoś” pobral kilkadziesiąt tysięcy złotych z bezosobowego funduszu plac. Nie chcemy, aby nas w ten sposób wyreczono w sprzedaży usług, w których się specjalizujemy.

Sprawa ma także inny aspekt. Gdybyśmy puścili „na żywioł” rozwój naszych systemów i „zdekonspirowali” je w zakresie struktury zbiorów, oprogramowania itd., to w wyniku niekontrolowanych przez nas modyfikacji powstałyby nowe wersje systemów, naszych systemów, dla których nie moglibyśmy już świadczyć usług konserwacyjnych i ponosić odpowiedzialności za ich jakość. Dla zapewnienia więc jednolitości systemu powtarzalnego we wszystkich zastosowaniach nie możemy dopuścić do dowolnych zmian i modyfikacji, zwłaszcza u tych słabszych, mniejszych użytkowników, którzy w wypadku awarii systemu zwrócą się do nas o pomoc.

Zrozumiałe, że inna jest sytuacja wielkich użytkowników, dysponujących silnymi ośrodkami obliczeniowymi, które mogą być partnerami dla ZETO. Oni mają inne podejście do kupowanego systemu. Chcieliby znać wszystkie tajniki i reguły dokumentacji projektowej. Oczywiście nie dla nich są przeznaczone nasze systemy powtarzalne. Ale jeśli chcą z nich korzystać, to kupują — na innych oczywiście warunkach finansowych, jak np. POLMO — całą dokumentację, i mogą z nią robić co chcą, a my z nimi chętnie kontynuujemy współpracę, na zasadzie kooperacji.

TEZA 3: Kolejne dwa problemy: a) dla kogo są przeznaczone będące obecnie w obiegu systemy powtarzalne i b) jaki jest zakres przedmiotowy ich zastosowań.

Dla kogo przeznaczone są systemy powtarzalne?

K. Szumlas: Należy sobie powiedzieć wyraźnie, że nie uszczęśliwimy wszystkich systemami powtarzalnymi. W moim rozumieniu są one przeznaczone przede wszystkim dla wielu tysięcy użytkowników, którzy w zakresie informatyzacji zarządzania nie są sterowani przez centralne ani branżowe ośrodki obliczeniowe, którzy są

zdani na własne sily i dla których zamawianie indywidualnego systemu informatycznego łącznie z jego zaprojektowaniem i oprogramowaniem byłoby finansowym samobójstwem, gdyż taki prosty, ale robiony „na miarę” system kosztuje dzisiaj 4—5 mln zł. Takich niewielkich zakładów i przedsiębiorstw jest wiele tysięcy. Niektóre z nich reflektują tylko na pewne odcinkowe podsystemy, mające usprawnić określone agendy działalności przedsiębiorstwa. Inną grupą naszych klientów są komórki ETO zatrudniające po kilka osób, które nie mają komputera, nie mają też projektantów systemów i programistów. Są to organizatorzy rozwoju informatyki w swoim przedsiębiorstwie lub nawet w swojej branży, na przykład w organizacji SPOLEM. Dla nas produkcja i rozpowszechnianie systemów powtarzalnych dla tego właśnie typu klientów jest podstawowym zajęciem. Są także zjednoczenia grupujące dziesiątki przedsiębiorstw, oczekujących na gotowe, powtarzalne systemy informatyczne. Nawiazaliśmy dotąd współpracę z 13 takimi zjednoczeniami, co oznacza świadczenie usług dla ponad setki konkretnych klientów.

Wśród zbiorowych użytkowników są też tacy, którzy właściwie naszych usług nie potrzebują, gdyż mają własne silne ośrodki obliczeniowe. Ale takim jednostkom oferujemy — tylko na innych już warunkach — nasze systemy powtarzalne z pełną dokumentacją i wchodzimy z nimi w kooperację w zakresie rozwoju systemów.

Zakres przedmiotowy zastosowań systemów powtarzalnych

K. Szumlas: Systemy powtarzalne mają zawsze charakter odcinkowy, a więc ograniczony zakres zastosowań. Jeśli produkowane są w jednym ośrodku, np. w ZETO-Bydgoszcz, to są zintegrowane na poziomie kodów, a nawet niekiedy na poziomie zbiorów. Ale generalnie są przewidziane do zastosowań ewidencyjnych. Robiliśmy próby zbudowania powtarzalnego systemu o charakterze decyzyjnym w dziedzinie planowania. Udało się nam wdrożyć ten system u 4 użytkowników. Nie uzasadnia to jednak powtarzalności takiego systemu i nie wynikają z tego dla nas żadne korzyści finansowe. Po prostu zachodzą poważne różnice między zastosowaniami o charakterze ewidencyjnym i decyzyjnym.

Zastosowania informatyki w dziedzinie ewidencji odznaczają się tym, że: 1) dotyczą przeszłości, 2) wykorzystywane są dla ustalenia stanu prawnego i odpowiedzialności, 3) nie zależą od konkretnej organizacji, lecz głównie od ogólnych przepisów.

Natomiast zastosowania decyzyjne dotyczą przyszłości, są źródłem określonego działania i zależą od różnorodnych organizacji. Przy istniejących u nas rozmaitych strukturach organizacyjnych, w których nawet w jednej branży trudno nieraz znaleźć wspólne struktury organizacyjne w różnych

przedsiębiorstwach, opracowanie powtarzalnego systemu o charakterze decyzyjnym wydaje mi się po prostu niewykonalne. Nie wyobrażam sobie takiego systemu działającego w obszarach decyzyjnych, który można by dostosować bez zasadniczych, gruntownych przeróbek do każdej organizacji gospodarczej.

Jak wynika z przeprowadzonej dyskusji systemy indywidualne należą już do przeszłości, natomiast powtarzalne systemy informatyczne przez wielu swoich walorach — cena, jakość, okres adaptacji i wdrożenia — mają określone ograniczenia dotyczące:

- udostępniania użytkownikowi przede wszystkim tylko dokumentacji eksploatacyjnej

- obszaru zastosowań — wyłącznie zagadnienia ewidencyjne

- przeznaczenia — głównie dla instytucji mniejszych i początkujących w zastosowaniu informatyki.

TEZA 4: Dalszy rozwój informatyki w Polsce nie może na tym poprzestać

L. Sankowski: Duża część naszej dyskusji obracała się wokół pojęcia systemów powtarzalnych, co odzwierciedla stan naszej świadomości i stopień opanowania tej problematyki. Natomiast mniej było głosów dotyczących następnej generacji, tzw. oprogramowania narzędziowego, pozwalającego automatyzować proces projektowania i oprogramowania systemów informatycznych. Wiele tych barier, o których mówiliśmy w odniesieniu do systemów powtarzalnych, a w szczególności ograniczenie obszaru zastosowań do zagadnień ewidencyjnych i problem tzw. „czarnej skrzynki” — przy zastosowaniu oprogramowania narzędziowego nie istnieje. Jeśli prześledzimy te sprawy np. na przykładzie pakietu generacyjnego STEP, to stwierdzimy, że zastosowania dotyczą zagadnień decyzyjnych, zaś dokumentacja oprogramowania jest elementem jawnym. Użytkownik uzyskuje tu program źródłowy, w który może dowolnie ingerować, zmieniać go, może i powinien go odpowiednio adaptować. W systemie powtarzalnym jesteśmy cały czas skazani na różnego rodzaju „doróbki” i „przeróbki” w trybie indywidualnym.

T. Zdziarski: Z pewnością nowe problemy pojawiają się przy szerszym, masowym opracowaniu nie indywidualnych systemów o dziedziny planowania. Systemy te są uzależnione od specyficznych cech funkcjonowania poszczególnych branż, a nawet przedsiębiorstw. Dla tej dziedziny trzeba będzie zastosować systemy odpowiadające zasadniczo innym regulom niż systemy powtarzalne typu ewidencyjne-go.

K. Szumlas: Systemy powtarzalne, jakkolwiek są ważną drogą, wiodącą do zmniejszenia społecznych kosztów rozwoju informatyki, nie są oczywiście drogą jedyną... Mówiłem o ograniczeniach systemów powtarzalnych przeznaczonych głównie dla mniejszych jednostek gospodarczych i dla początku-

jących w zastosowaniu informatyki. Inaczej wygląda sprawa oprogramowania narzędziowego czy technologicznego, którego producentem jest OBRI. Oprogramowanie, takie jak RODAN lub STEP, jest ważne choćby dlatego, że dystrybucja jego powinna przebiegać na zupełnie innych zasadach i w innych warunkach niż oprogramowania powtarzalnego. Osobiście sądzę, że oprogramowanie takie powinno być sprzedawane razem z komputerem, tak jak w NRD.

E. Petryk: Oprogramowanie narzędziowe stanowi nową jakość, do której trzeba się przyzwyczaić i dostosować. Np. STEP nie jest gotowym systemem powtarzalnym, jest generatorem oprogramowania, czyli czymś co stawia się chyba nieco wyżej. Część użytkowników nie jest jeszcze przygotowana do wykorzystywania tego systemu, bo jest przyzwyczajona do otrzymywania czegoś gotowego, o konkretnych wymaganiach i ograniczeniach. Natomiast w STEP-ie problem jest tak postawiony, że za pomocą tego narzędzia użytkownik sam rozwiązuje swój problem. W związku z powyższym odpowiedzialność za właściwe sformułowanie problemu pozostawia się bezpośrednio użytkownikowi i efekt końcowy, czyli przydatność systemu dla przedsiębiorstwa, zależy od jego dojrzałości i fachowości.

Spotkałem się z postulatami, aby w toku prezentowania pakietu narzędziowego przedstawiać klientowi konkretne tabulogramy potrzebne danemu użytkownikowi. Tu ujawnia się niezrozumienie funkcji oprogramowania narzędziowego, pomieszania pojęć ze znanymi systemami indywidualnymi i powtarzalnymi.

Zagadnienie jawności dokumentacji przedstawia się w tych pakietach zupełnie odmiennie niż w systemach powtarzalnych. Jawność całej dokumentacji jest korzystna także dla producenta oraz dla komórki zajmującej się sprzedażą systemu. Dostarczając użytkownikowi informacje o budowie zbiorów, a w przypadku systemu STEP także programy źródłowe, dajemy mu możliwość nie tylko poprawiania błędów, ale przede wszystkim wygenerowania tego, co mu jest potrzebne. W toku konsultacji staramy się wskazać, co tu można zmienić, co uzyskać. Wersja systemu STEP-75 może służyć do wygenerowania dowolnego systemu opartego na bazie danych. STEP-76 realizuje funkcje planowania i sterowania produkcją.

J. Bartczak: Jeżeli zakład eksploatuje wiele różnych i nie powiązanych ze sobą systemów, pojawiają się poważne trudności. W miarę rozszerzania obszarów informacyjnych podlegających automatyzacji nieodzowne staje się integrowanie systemów cząstkowych w uzupełniające się systemy duże, oparte np. na RODANIE. Chciałbym zwrócić uwagę na to, że w większych ośrodkach obliczeniowych generowane są własne systemy. Dobrze byłoby więc, aby ZETO miały na uwadze także kontakty z tymi użytkownikami, którzy w oparciu o oprogramowanie narzędziowe opracowali i eksploatują już własne systemy użytkowe.

TEZA 5: Rozpowszechnienie i zastosowanie nowoczesnego oprogramowania narzędziowego znajduje się jeszcze w początkowym stadium. Dalszy rozwój zależy od zastosowania właściwych metod i technik obrotu oprogramowaniem. O takich technikach mówiono w dyskusji, a mianowicie: o informacji, aplikacji, konsultacji.

Informacja

E. Stencel: Najważniejsza w rozpowszechnieniu jest dokumentacja informacyjna, która ma przekonać użytkownika, że dany system jest wart zainteresowania i zakupienia. Dokumentację tę wspieramy w ELWRO prezentacjami. Zapraszamy klientów tam, gdzie ten system jest eksploatowany i o systemie mówią im nie informatycy ale sami użytkownicy... Jednak informacji o systemach, o oprogramowaniu jest bardzo mało. Czy ktoś z Państwa wie, że u nas można kupić pakiet DMS łącznie z wdrożeniami? OBRI jest zobowiązany do prowadzenia ewidencji wszystkich istniejących programów i systemów. Robi to w sposób co najmniej niezadowolający.

Aplikacja

E. Stencel: Nawiązaliśmy współpracę ze Zjednoczeniem Informatyki, dotyczącą dostawy pakietu STEP dla konkretnego użytkownika na Śląsku. I co się okazuje? Sieć ZETO nie bardzo jest skłonna do pełnej obsługi użytkownika. Dostarczyć STEP — owszem, przeszkolić — także, ale zaprojektować konkretny system użytkowy w oparciu o pakiet STEP — to nie. Dlaczego. Przecież ci, którzy przeszli 2-tygodniowe szkolenie w Łodzi, nie są jeszcze przygotowani tak dobrze, aby samodzielnie zaprojektować system użytkowy w oparciu o narzędzie programowe. Oczywiście zgadzam się z potrzebą bardzo szerokiego udziału użytkownika przy opracowywaniu systemów użytkowych, generowanych z oprogramowania narzędziowego, ale użytkownikowi trzeba przy aplikacji systemu narzędziowego konkretnie pomagać.

Konsultacja

E. Petryk: Podczas obrotu oprogramowaniem narzędziowym nie można zastępować użytkownika przy generowa-

niu konkretnych systemów użytkowych, ale należy go wspierać poprzez doradztwo informatyczne i konsultacje. Jeśli np. jakiś kombinat zakupi STEP i będzie go wykorzystywał w przedsiębiorstwach filialnych, to w istocie rozpowszechnienie tego systemu będzie ograniczone przez doradztwo. Konsultacje i serwis informatyczny otrzymuje tylko ten, kto pakiet kupił. Jeśli zaś kombinat chciałby rozpowszechnić oprogramowanie wśród podległych przedsiębiorstw, to musi podjąć się pilotowania wynikających z tego problemów, powinien wziąć na swe barki ciężar konsultacji i doradztwa. Nie jest to proste i nie każdy zespół jest do tego przygotowany.

R. Myśliwiec: Wiele omówionych w dyskusji spraw wymaga dalszych działań i decyzji. Mam nadzieję, że wymiana poglądów przyczyni się do bardziej efektywnej współpracy producentów systemów i ich użytkowników.

Opracował Artur HAJNICEZ

Zdjęcia Andrzej KLIMEK

Krajowa konferencja DIEBOLDA

Na miejsce obrad kolejnej krajowej konferencji organizowanej cyklicznie w ramach współpracy z Europejskim Programem Badawczym DIEBOLDA wybrano tym razem Toruń. Stąd obok organizatora „z urzędu”, jakim jest OBRI, współgospodarzem konferencji zostało bydgoskie ZETO, któremu dzięki uprzejmości toruńskich Zakładów ELANA udało się uzyskać wyjątkowo atrakcyjne miejsce obrad. Była nim siedziba Klubu Zakładowego tego przedsiębiorstwa, mieszcząca się w pomysłowo zaadaptowanym XVII-wiecznym magazynie handlowym.

W odróżnieniu od poprzednich konferencji, organizatorzy znacznie ograniczyli liczbę uczestników, których do Torunia przybyło łącznie ok. 35. Zgodnie z tradycją, dwudniowy program konferencji (6 i 7 listopada ub.r.) obejmował referat przedstawiciela EPB DIEBOLDA, referaty sprawozdawcze z wiosennej międzynarodowej konferencji w Wenecji, referaty na temat osiągnięć krajowych oraz posiedzenie polskiej grupy doradczej ds. współpracy z EPB DIEBOLDA.

Jak zwykle z największym zainteresowaniem oczekiwano na wystąpienie gościa zagranicznego, którym tym razem był dyrektor Programu, p. Vaucher. Niestety treść jego referatu pt. „Nowe tendencje w technologii przetwarzania danych”, koncentrująca się głównie na bardzo popularnym przedstawieniu przemian i prognoz rozwoju sprzętu komputerowego, niewątpliwie rozczarowała większość słuchaczy, którzy i tak na bieżąco śledzą ten rozwój bezpośrednio z lektury prasy zagranicznej. Jedynym bardziej interesującym punktem tego referatu była prognoza trendów rozwojowych informatyki do 1985 r., wyrażona w zmianach struktury kosztów przetwarzania. Wg tej prognozy udział kosztów osobowych zwiększy się z obecnych 40 do ok. 80%, głównie w wyniku radykalnego zmniejszenia się udziału kosztów sprzętu (z 40 do ok. 6%). Zmianom tym towarzyszyć będzie stopniowe zmniejszanie się dominującej roli komórek informatycznych wewnątrz przedsiębiorstw w wyniku przejmowania coraz większej liczby funkcji przetwarzania przez bezpośrednich użytkowników. Na zapytanie

jednego z uczestników, czy też ostatni trend oznacza również ograniczenie roli usług informatycznych, p. Vaucher odpowiedział, że usługi te zmieniają jedynie swój charakter, ponieważ ich punkt ciężkości przesunął się na działalność szkoleniowo-doradczą, głównie w fazie wdrożeń.

W kolejnym referacie dr W. Staniszkis zwięźle scharakteryzował podstawowe parametry techniczne i użytkowe, dotychczas poniesione nakłady oraz aktualny stan wdrożeń i orientacyjne koszty implementacji systemu zarządzania bazą danych RODAN. Podobny charakter miał referat mgr. inż. E. Cyrkłafta pt. „Pakiet programowy DMS-2 jako narzędzie budowy modułowego systemu kierowania obiektem przemysłowym MOSKOP”, prezentujący interesujące osiągnięcia ZETO-Bydgoszcz.

Pierwszy dzień konferencji zamknął referat sprawozdawczy dyr. B. Gliksmana z obrad XL konferencji EPB DIEBOLDA w Wenecji (18—20 kwietnia 1978 r.) na temat trendów rozwojowych sprzętu, a więc pokrywający się w swej zasadniczej treści z wywodami p. Vauchera, który główne akcenty swego referatu musiał czerpać z materiałów tej właśnie konferencji. W tej sytuacji drugi referat sprawozdawczy z konferencji weneckiej, dotyczący trendów rozwojowych oprogramowania, musiał wywołać znacznie większe zainteresowanie, tym bardziej, że jego treść była bardziej zbieżna z profilem zainteresowań zawodowych większości słuchaczy. Referat ten, wygłoszony w pierwszej części drugiego dnia obrad przez dyr. J. Golińskiego, wyróżniał się doskonałym sposobem prezentacji, ilustrowanej bardzo czytelnymi planszami graficznymi.

Drugą część dnia zarezerwowano na wspomniane już posiedzenie Grupy Doradczej. Jego program obejmował wstępną ocenę obecnej konferencji, informacje o kolejnych europejskich konferencjach DIEBOLDA w roku przyszłym oraz ustalenie terminów i miejsca kolejnych konferencji krajowych. W odniesieniu do tych ostatnich postanowiono utrzymać zasadę ograniczania liczby uczestników do ok. 50 osób.

Władysław KLEPACZ

Programowanie modułowe w praktyce

Dziesięcioletnie doświadczenia związane ze stosowaniem komputera II generacji w czeskosłowackim przedsiębiorstwie budownictwa przemysłowego TEPLOTECHNA pozwoliły uzyskać na komputerze III generacji — R-32 — lepsze efekty eksploatacyjne, niż formalnie umożliwiają to jego nowoczesne rozwiązania sprzętowe i oprogramowanie firmowe. Komputer ten ukierunkowany na pamięci dyskowe oraz wyposażony w nowoczesne oprogramowanie podstawowe wymagał zasadniczej reorientacji w projektowaniu zastosowań, a zwłaszcza w tworzeniu oprogramowania użytkowego.

Nowe podejście należało oprzeć przede wszystkim na udogodnieniach wynikających z zastosowania metody programowania modułowego. Przy tworzeniu oprogramowania modułowego dla nowego problemu komputery R-32 — i inne maszyny III generacji — pozwalają uniknąć opracowywania od podstaw wszystkich elementów tego oprogramowania. Powstaje wówczas możliwość wykorzystania elementów (modułów) poprzednio wykonanych własnych programów użytkowych lub oprogramowania firmowego.

Podstawowym celem niniejszego artykułu jest pokazanie sposobu wykorzystania metody programowania modułowego i jego niewątpliwych zalet w praktycznym zastosowaniu komputera R-32 w Czechosłowacji. Jako przykład wybrano podsystem operatywny ewidencji środków trwałych oraz element tego podsystemu, dotyczący opłat za korzystanie z tych środków w produkcji budowlanej.

W projekcie wspomnianego podsystemu niezbędne oprogramowanie użytkowe podzielono na 4 bloki (zespoły) programów, ściśle precyzując ich zadania. Bloki te, oznaczone literami A, B, C i D, objęły łącznie 17 modułów, a mianowicie:

- blok A o 3 modułach, realizujących wczytanie kart dziurkowanych i ich konwersję na taśmę magnetyczną oraz sortowanie i aktualizację ewidencji podstawowej środków trwałych
- blok B o 4 modułach, realizujących dwa rodzaje sortowania oraz wydawnictwa wynikowe opracowywane w cyklu miesięcznym
- blok C o 4 modułach, realizujących dwa rodzaje sortowania oraz wydawnictwa wyników opracowywane w cyklu kwartalnym
- blok D o 6 modułach, realizujących trzy rodzaje sortowania oraz wydawnictwa wynikowe opracowywane w cyklu półrocznym i rocznym.

Przy stosowaniu tradycyjnych metod programowania rozdział zadań powodował istotne trudności ponieważ złożoność i zakres poszczególnych programów były bardzo różne.

Stosując metodę programowania modułowego powstała możliwość takiego podziału pracy aby stopień trudności realizacji jednego bloku programów oraz jego modułów odpowiadał zarówno możliwościom, jak i wydajności programistów wyznaczonych do realizacji tego zadania. Poprzednie doświadczenia dowiodły również, że cały program jako samodzielne zadanie w większości przypadków przekracza możliwości efektywnego rozwiązania przez jednego programistę, a także nie stwarza możliwości skutecznej bieżącej kontroli wykonywanych prac.

Podzieliwszy oprogramowania na bloki przeprowadzono kolejny podział — programów na moduły, a następnie opracowano całościowy schemat blokowy podsystemu. Schemat ten korygowano w miarę pojawiających się w toku opracowywania bieżących potrzeb modularyzacji, ustalając dokładnie sposób wzajemnego komunikowania się poszczególnych modułów. Zakres funkcjonalny każdego modułu ograniczano w taki sposób, aby można było go jednoznacznie identyfikować oraz bez szczegółowej analizy rozumieć jego wszystkie funkcje. Założono również, że każdy moduł powinien umożliwiać szybką jego modyfikację i to w taki sposób, aby dokonane zmiany nie naruszyły ustalonego zakresu działania innych modułów programu. Nie dotyczyło to oczywiście przypadków wprowadzania zasad-

niczych zmian typu funkcjonalnego, dotyczących z reguły większej liczby modułów.

Zgodnie z tymi założeniami każdy program został podzielony na moduły stosunkowo małe i proste pod względem funkcjonalnym, obejmujące z reguły realizację tylko jednej funkcji. Położono również duży nacisk na maksymalne uproszczenie sposobu rozwiązania wzajemnej komunikacji modułów wewnątrz programu.

Dzięki takiej modularyzacji programów stwierdzono, że nawet w najbardziej złożonych zadaniach znaczna część modułów ma rzeczywiście prostą konstrukcję i dlatego ich opracowanie można zlecać nawet początkującym programistom.

Podział całego problemu na zadania cząstkowe, którym odpowiadały poszczególne moduły, powierzono do wykonania programiście najbardziej doświadczonemu. Opracowanie programów bardziej obszernych dzielono w zespole wykonawczym pomiędzy kilku programistów o różnicowanych kwalifikacjach, spośród których najbardziej doświadczony był odpowiedzialny za działanie całego programu. Programista ten rozstrzygał zwłaszcza o szczegółach dotyczących sposobu ujednoczenia komunikacji pomiędzy modułami.

Testowanie różnych programów modułowych (i modułów) odbywało się równolegle. W wielu przypadkach stosowano w tym celu symulację interfejsów innych, nie uruchomionych jeszcze modułów. Dla każdego testowanego modułu lub ich zespołu główny programista tworzył prosty moduł sterujący, który wymagał tylko wypełnienia pól niezbędnych do funkcjonowania testowanego modułu oraz wydruku jego informacji wynikowych. Po przetestowaniu poszczególnych modułów następowało oczywiście testowanie całego programu.

Przy modularnym sposobie programowania osiągnięto zdecydowanie lepsze warunki kontroli prac, ponieważ rozmiary poszczególnych modułów ustalono tak, aby całkowita pracochłonność przygotowania jednego modułu nie przekraczała 4 dni pracy, rozłożona oczywiście na dłuższy okres ze względu na praktyczne możliwości testowania.

Doświadczenia uzyskane podczas realizacji podsystemu wykazały, że niezbędne operatywne zmiany w rozwiązywaniu organizacyjnym problemu, które powodowały również zmiany w programach, przy programowaniu modułowym można było przeprowadzić znacznie szybciej, ponieważ w większości przypadków sprowadzały się one do zmiany zaledwie jednego, a maksymalnie kilku modułów. Szybsze i łatwiejsze było również wykrywanie błędów programowania ze względu na dogodniejsze warunki ich lokalizacji w wyniku wspomnianego ograniczenia rozmiarów modułów.

Modularyzacja pozwoliła również zaoszczędzić czas komputera R-32, potrzebny do uruchamiania programów, ponieważ znacznie odtańd sprawniejsze uruchamianie małych modułów trwało w sumie krócej niż uruchamianie tradycyjnych programów monolitycznych.

Warto wreszcie wspomnieć, że modularyzacja umożliwia także opracowywanie poszczególnych modułów tworzących jeden program w różnych językach programowania, co w istotny sposób ułatwiło podział zadań oraz organizację wewnątrz zespołów programistów.

Celem lepszego wykorzystania istniejącego oprogramowania podstawowego R-32 niezbędne było również opracowanie specjalnych modułów służących do komunikowania się całych programów lub ich modułów z systemem operacyjnym oraz ze zbiorami danych.

Nie wchodząc w szczegóły techniczne opisanego zastosowania metody programowania modułowego, chciałbym raz jeszcze podkreślić, że doświadczenia zebrane w przedsiębiorstwie TEPLOTECHNA potwierdziły bezsporne zalety tej metody.

Jiří ZAVESKÝ
TEPLOTECHNA
Praha (CSRS)

Wedle stawu grobla

Jest ich w Polsce kilkaset, a może więcej. Policzyć dokładnie da się tylko te, które mają własne komputery, ale większość ich nie ma. Duże — na miarę zakładu, dla którego powstały — mają z reguły efektowny lokal, importowany sprzęt, sporo ciekawych zadań. Dobrze są znane w środowisku informatyków. Małe, zakładowe ośrodki informatyki nie mają takiej sławy.

Zakładowy Ośrodek Przetwarzania Informacji w Zaodrzańskich Zakładach Przemysłu Metalowego im. M. Nowotki w Zielonej Górze mocno jest wrośnięty w strukturę organizacyjną macierzystego zakładu. Właśnie obchodzi dziesięciolecie. W ciągu tych dziesięciu lat dobrze zasłużył się ZASTALOWI. Powstał z ewidentnej potrzeby.

Brakowało rąk do pracy. Ludzi z administracji przesuвано do bezpośredniej produkcji. Zielona Góra metropolią przecież nie była, a liczba mieszkańców rosła wolniej niż przybywało stanowisk pracy. Przesunięcia w obrębie 3-tysięcznej załogi Zakładów były możliwe tylko pod warunkiem automatyzacji prac administracyjnych. Powstały więc warunki rozwoju informatyki, informatyki jednak nie było... Nie licząc ZETO, które wprawdzie powstało nieco wcześniej niż ZOPI ZASTALU, lecz oprócz planów nie miało właściwie nic.

Trudno dziś rozstrzygnąć jak to właściwie było, ale wydaje się, że ZASTAL był wówczas w stanie kupić sobie własny komputer. Ostatecznie przyjęto jednak bardziej ekonomiczną koncepcję komputeryzacji i czekano na rozwój ZETO. Uznano bowiem, że: ● ZOPI nieprędko potrafi obciążać komputer w sposób opłacalny

● ZETO musi stosunkowo szybko otrzymać komputer

● ZETO będzie potrzebowało i poszukiwało klientów w pierwszej kolejności w grupie dużych przedsiębiorstw.

I jak się okazało taki sposób rozumowania był prawidłowy. Wprawdzie poskromienie apetytów na własny sprzęt oraz związanie swych losów z miejscowym ZETO może nieco opóźniło automatyzację procesów zarządzania w ZASTALU, ale w roku 1978 ZOPI nie miało kłopotów z rentownością zastosowania informatyki. Przyjęcie natomiast 10 lat temu rozwiązania alternatywnego dziś oznaczałoby dotkliwy kłopot ze sprzedażą prawie dwóch zmian pracy komputera.

Faktem jest, że i ZETO przechodziło niełatwe koleje. Liczne więc słabości ZETO bezpośrednio oddziaływały na wyniki działalności ZOPI: początkowo bardzo słaby sprzęt (przez długi czas ODRA 1003), potem transport zadań do Wrocławia i — odwrotnie — wyników, kadra bez wystarczającej praktyki, kłopoty lokalowe.

Trzeba przyznać, że ZETO dość szybko okrzepło, a stosunki partnerskie, które od początku ułożyły się dobrze, z czasem umocniły się jeszcze bardziej w wyniku wzajemnie świadczonej pomocy.

Borykające się z kłopotami lokalowymi ZETO wsparło się bowiem na możliwościach ZASTALU. W 1977 roku zainstalowano ODRE 1305 na terenie ZASTALU, który użyczył 270 m² odpowiednio adaptowanej powierzchni. Za cenę tej przysługi ZOPI uzyskało priorytet w korzystaniu z komputera.

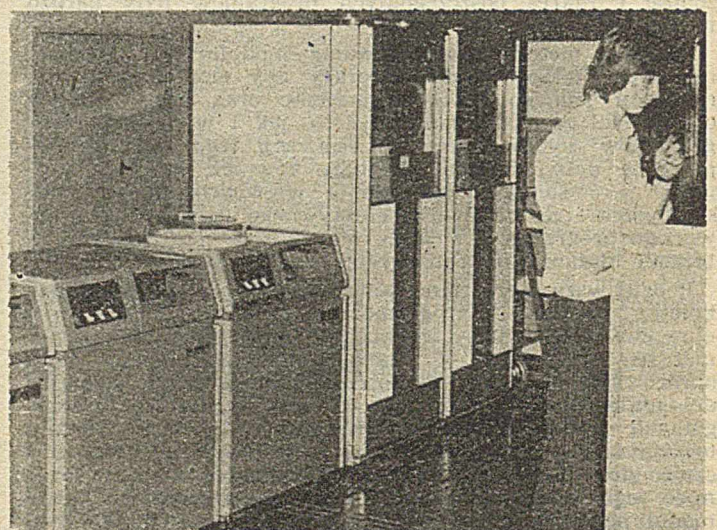
W 1977 roku ZOPI wykorzystało ponad 3 tysiące godzin pracy ODRE 1305. Był to efekt eksploatacji systemów zarówno przeniesionych z wrocławskiej ODRE 1304, jak i nowych systemów. 1100 godzin absorbowało obliczanie płac, 550 — gospodarka materiałowa, 350 — techniczne przygotowanie produkcji, 70 — normatywny rachunek kosztów.

Można powiedzieć, że to banalne zastosowania. Ale dla ZASTALU prace te mają istotne znaczenie. Zanim będzie można stosować systemy o większym ciężarze gatunkowym wymienione prace pozwoliły już zmienić strukturę zatrudnienia w zakładzie. Radykalnie zmieniła się proporcja pomiędzy grupą pracowników bezpośrednio-produkcyjnych i administracji w postulowanym 15 lat temu kierunku (obecnie kształtuje się jak 4:1).

Rachunek ekonomiczny i bilans możliwości wyznaczał kierunki prac projektowo-programowych, a więc tego, co korzystnie można było przejąć lub zakupić od innych, nie wykonywano własnymi siłami. Jest więc w ZOPI oprogramowanie „made in TASKO”, „made in IINTE”, lub „made in ZETO”. Jest też sporo oprogramowania własnego, które jest o tyle lepsze, że powstaje w ścisłej konsultacji z użytkownikiem bezpośrednim. Inna rzecz, że do prac nad oprogramowaniem delegowano tylko 11 osób, a przecież większość czasu pochłania konserwacja systemów eksploatacyjnych. W najbliższym czasie zespół ten będzie w znacznym stopniu obciążony przenoszeniem istniejących zbiorów i systemów na dyski (w ZETO zainstalowano 4 jednostki dyskowe po 7.25 MB).



Dzięki zainstalowaniu ekranowych urządzeń końcowych, Ośrodek zaoszczędzi karty, zaś operatorkom oszczędzi się hałasu



Dyski to nie jedyne novum w ośrodku ZETO na terenie ZASTALU
Zdjęcia: Stefan GRUSZEWSKI

Dyski to nie jedyne novum w ośrodku, z którego korzysta ZOPI. Do usprawnienia procesu przygotowania danych ZASTAL zakupił dla ZOPI 8-stanowiskowe urządzenie typu VTS 56 100 do wprowadzania danych w trybie *on-line*. Usprawni to i ułatwi pracę operatorom, a także umożliwi zdalne testowanie programów.

Tak więc do pełnej automatyzacji ZASTALU ZOPI zmierza metodą indukcyjną: do sprawdzonych rozwiązań dodaje nowe jakości. Do rosnących potrzeb dopasowuje sprzęt, a do możliwości tego sprzętu doszkała niezbędną kadre.

Na razie — jak deklaruje ZOPI — własny komputer jest niepotrzebny. Ponieważ współpraca z ZETO w dalszym ciągu układa się bardzo dobrze (szczególnie pomocni są dyrektorzy —

mgr M. Bencych i inż. Z. Grajczak) i gwarantuje możliwość zwiększenia czasu korzystania z komputera. Wydaje się, że takie status quo jest korzystne dla obu stron.

Być może, że prezentacja Zakładowego Ośrodka Przetwarzania Informacji w ZASTALU nie wypadła błyskotliwie. Rzecz w tym, że błyskotliwie wypaść nie powinna. Chodzi tu bowiem o zilustrowanie upowszechniającej się od niedawna nowej idei zastosowań informatyki, a mianowicie racjonalnej proporcji między rzeczywistą wielkością potrzeb a wielkością ośrodka obliczeniowego.

Pozornie ZOPI miało warunki na rozwiązanie efektywne, których sobie nie skąpią liczne inne ośrodki. Bo przecież tak renomowany zakład, jakim jest ZASTAL, zatrudniający 3 ty-

siące pracowników i produkujący rocznie za 6 mld złotych sprzęt porządany w kraju i za granicą (56% produkcji na eksport), niewątpliwie stać było na własny komputer. Zwłaszcza że zarówno naczelny dyrektor, inż. Aleksander Tworowski, jak i jego zastępca ds. ekonomicznych, mgr Czesław Gendera, są gorącymi orędownikami wdrożenia informatyki w procesy zarządzania ZASTALEM. Wybór rozwiązania w myśl starego porzekadła „wedle stawu — grobla” wystawia więc dobre świadectwo kierownictwu ZOPI i kierownictwu ZASTALU.

Tekst: Krystyn BERNATOWICZ

Czy „branżowy” stanie się „centralnym”?

Z biuletynu Resortowego Ośrodka Badawczo-Rozwojowego Informatyki Rolnictwa¹⁾: *Po blisko 15 latach ewolucyjnego rozwoju elektronicznej techniki obliczeniowej wychodzimy już z zakresu eksperymentowania (...). Zapoczątkowujemy (...) pomoc dla służb rolniczych i administracji różnych przedsiębiorstw resortu w dziedzinie stosowania ETO. Widzimy potrzebę ułatwiania wymiany doświadczeń (...), informowania o osiągnięciach praktycznych, a nawet prezentowania konkretnych rozwiązań organizacyjno-technicznych, przydatnych do szerszej adaptacji w warunkach gospodarki rolnej o swoistej przecież specyfice produkcyjnej.*

Dalej²⁾ podane są informacje o potencjale obliczeniowym, który zgromadził resort na potrzeby rolnictwa. Nie było tego za wiele:

- w resorcie działało 25 ośrodków informatyki dysponujących własnym sprzętem
- zatrudniono w nich 600 osób
- dysponowano 9 komputerami ODRA (5 komputerów 1305, 2 komputery 1325, 1 komputer 1304, 1 komputer 1204) i 2 komputerami ZAM-41 oraz 31 minikomputerami MERA 300, minikomputerem RC 5500 i K 202. W ubiegłym roku wykonane usługi wyrażały się wartością 111 mln zł.

Wyniki nie były imponujące. Przerób tego rządu osiąga jeden zakład eto z sieci Zjednoczenia Informatyki.

Na każdy komputer przypada nieco więcej niż 10 mln zł³⁾, a pozwala to przypuszczać, że średni czas ich pracy odpowiada jednej zmianie.

W dziedzinie oprogramowania użytkowego również nie zanotowano rewelacji. Dokonana inwentaryzacja wykazała, że w resorcie eksploatuje się ok. 20 systemów, 40 podsystemów i kilkadziesiąt pakietów programów.

W cytowanym już opracowaniu zasługują na uwagę dwa inne momenty. Pierwszy, który dotyczy potrzeby koordynacji i powielarności oprogramowania. Drugi, w którym komunikuje się, że najsilniejszy kadrowo (160 osób) i najlepiej wyposażony w sprzęt (2 komputery ODRA 1300) jest Ośrodek Centralnego Zarządu Budownictwa Rolniczego — BETOBR w Pruszczy koło Gdańska. Wybrałem się więc do tego ośrodka.

Parę lat temu resort rolnictwa podpisał umowę ze Zjednoczeniem Informatyki. Nie jest zamiarem autora mieć obopólnie korzystną współpracę. Wydaje się jednak, że wspierając się o duży potencjał obliczeniowy sieci Zjednoczenia Informatyki, nie byłoby rozsądnie zaniechać idei utworzenia własnego centralnego ośrodka, który z większą znajomością rzeczy mógłby się zająć opracowywaniem i

testowaniem uniwersalnych systemów informatycznych na potrzeby jednostek gospodarczych resortu. Czy mógłby nim być ośrodek, którego nie trzeba tworzyć od podstaw — a więc na przykład BETOBR?

Dorobku, który po pięciu latach działalności (październik 1978) podsumował BETOBR, nie powstydziliby się żaden ośrodek. Od początku swego istnienia ulokował się w bardzo przyzwoitych warunkach: dwupiętrowy, nowoczesny biurowiec na bliższych Gdańsku peryferiach Pruszcza zapewnia korzystanie z 2 800 m² powierzchni (w tym 200 m² sali komputera i 200 m² pomieszczeń socjalnych).

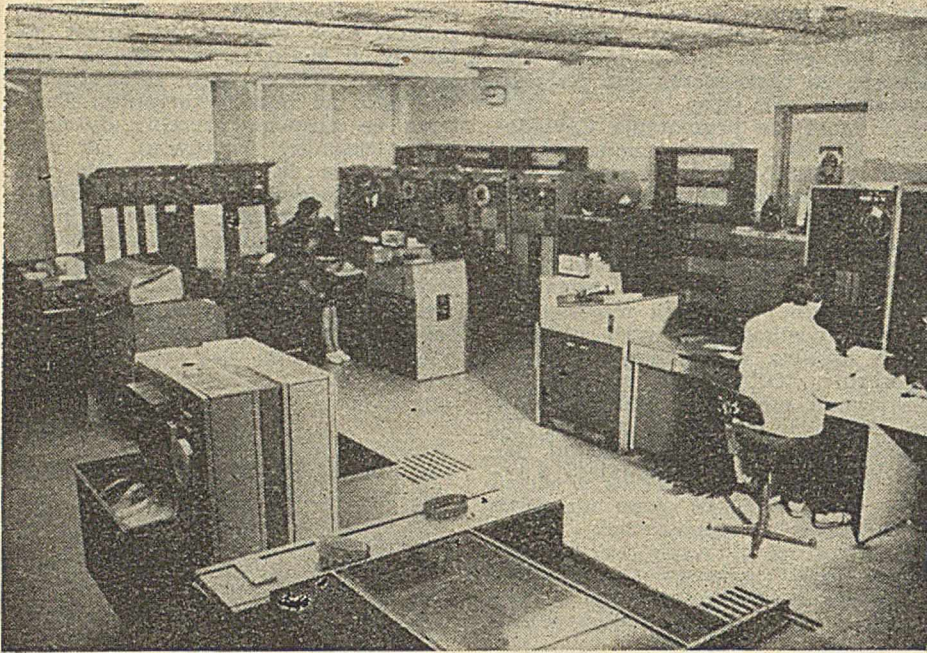
Wyznacznikiem możliwości BETOBRU jest ilość i jakość sprzętu. A więc przede wszystkim dwa komputery — zainstalowana w pierwszej kolejności (1973 r.) ODRA 1304 z pamięcią operacyjną 64 K, czterema jednostkami pamięci bębnowej i sześcioma — pamięci taśmowej oraz zainstalowana w połowie 1976 r. ODRA 1305 (PAO — 64 K, 6 jednostek pamięci taśmowej, 4 jednostki dyskowe po 7,25 MB). Słabością obu instalacji jest mała ilość urządzeń wejścia i wyjścia — w sumie 2 drukarki i 2 czytniki kart.

Jeżeli jednak skąpa ilość drukarek stanowi poważny problem, to z czytnikami sprawa wygląda znacznie lepiej. Funkcje czytników zostały zredukowane do minimum dzięki zainstalowaniu duńskiego systemu RC 3600 — 12-stanowiskowego urządzenia do rejestracji danych na taśmie magnetycznej. W ten sposób z kart korzysta się tylko incydentalnie.

3) Jest to spore uproszczenie jako, że wyliczone minikomputery są w stanie wykonać prace o wartości ekwiwalentnej dla możliwości 5 komputerów ODRA. Z uwagi na przypisanie minikomputerów do instytucji nie mających charakteru usługowego, można przypuszczać, że ich praca nie jest liczona w ogólnej wartości usług.

1) Informatyka w rolnictwie — Biuletyn ROBRIR, Warszawa czerwiec 1977.

2) Tamże, K. Łastowiecki: „Służby i prace informatyczne resortu”.



Sala komputerowa o powierzchni 200 m². Na łamach INFORMATYKI prezentowaliśmy już dwa razy mniejsze a podobnie wyposażone — np. sala gdańskiego ETOBU lub poznańskiego ZETO. Jeżeli jednak nastąpią zapowiadane w tekście uzupełnienia sprzętowo-powierzchnia będzie zbyt szczupła. Gospodarze ośrodka pocieszają się, że w najgorszym przypadku (w najlepszym wybuduje się drugą salę) system RC zostanie przesunięty do nieklimatyzowanej salki, zaś opróżniona (z klimatyzacją) będzie siedzibą RIADA

Natomiast niezwykle przydatnym urządzeniem wyjściowym jest pisak X-Y 1934/6 produkcji firmy ICL, używany przy wyprowadzaniu obliczeń inżynierskich. Pisze on i rysuje na kalce o formacie 76,5 cm × 60 cm z dokładnością 0,1 mm.

Trzecim gwarantem możliwości BETOBRU jest poziom kadry. Trzeba tu wspomnieć, że tradycje są dłuższe niż obecna nazwa i status ośrodka. Początki prac na rzecz budownictwa rolniczego sięgają 1967 roku. Dzisiejszy BETOBR był wtedy pracownią ETO w Biurze Projektów Budownictwa Wiejskiego i sporą grupę ludzi prze-

szkolił na maszynach licząco-analitycznych. Wielu z nich można spotkać teraz w sztandarowych ośrodkach wybrzeża, a i nie mało w północno-wschodniej Polsce. Ci, co zostali w BETOBRZE mają na swoim koncie poważne osiągnięcia. Są także bardzo przywiązani do macierzystego ośrodka, o czym świadczy niski wskaźnik fluktuacji — podlegają mu zresztą przeważnie operatorki urządzenia RC.

W 1978 roku wartość produkcji BETOBRU wyniosła 37,5 mln. zł. Oba komputery pracowały na 2,5 zmiany, RC 3600 — na 2 zmiany. W zasadzie obszarem działania BETOBRU były

przedsiębiorstwa budownictwa rolniczego i wodnego. Największym klientem było Gdańskie Zjednoczenie Budownictwa Rolniczego, w którego skład wchodzi 19 przedsiębiorstw — 16 z nich to systematyczni abonenci usług ośrodka. Na rzecz tych przedsiębiorstw BETOBR świadczy usługi projektowe i przetwarzanie danych.

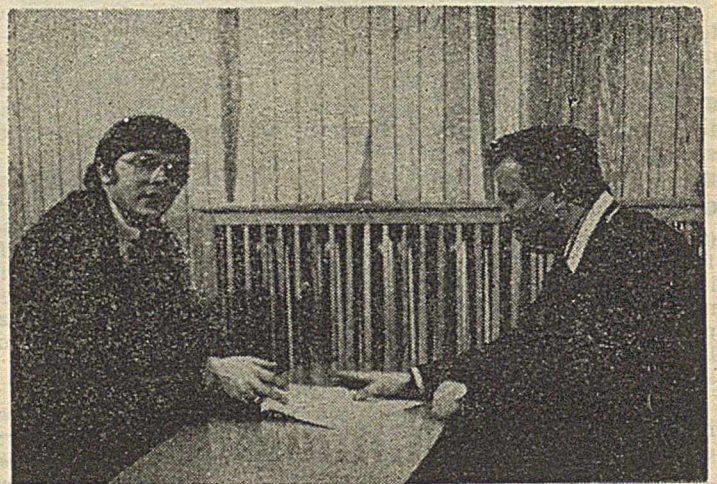
Zakres tematyczny prac projektowych jest szeroki, ale przede wszystkim ośrodek specjalizuje się w systemach gospodarki materiałowej. Koronnym jest tu SARGM-77 — uniwersalny system gospodarki materiałowej, który obecnie jest już nie tylko powszechny w budownictwie rolniczym, ale jest stosowany w całym resorcie rolnictwa. Innym, bardzo szeroko stosowanym systemem jest system kadrowo-zatrudnieniowy, natomiast klasycznym systemem centralnym jest AWIZO-MOC eksploatowany przez Centralny Zarząd Budownictwa Rolniczego i Centralny Zarząd Budownictwa Wodno-Melioracyjnego. System ten jak wiadomo dostarcza informacji o potrzebach inwestorów i zaangażowaniu mocy produkcyjnej wykonawców w procesie realizacji inwestycji. Na użytek przedsiębiorstw BETOBR opracował i utrzymuje jednolity indeks materiałowy, branżową tabelę norm, bazę normatywną dla systemu limitowania i normatywnego rozliczenia gospodarki materiałowej. W sumie opracowano 13 systemów i zespół programów do obliczeń inżynierskich. Są one eksploatowane w przedsiębiorstwach budownictwa rolniczego i wodno-melioracyjnego, w technicznej obsłudze rolnictwa, w państwowych gospodarstwach rolnych i spółdzielniach budownictwa wiejskiego.

W pięcioletnim okresie swojej działalności BETOBR wdrożył:

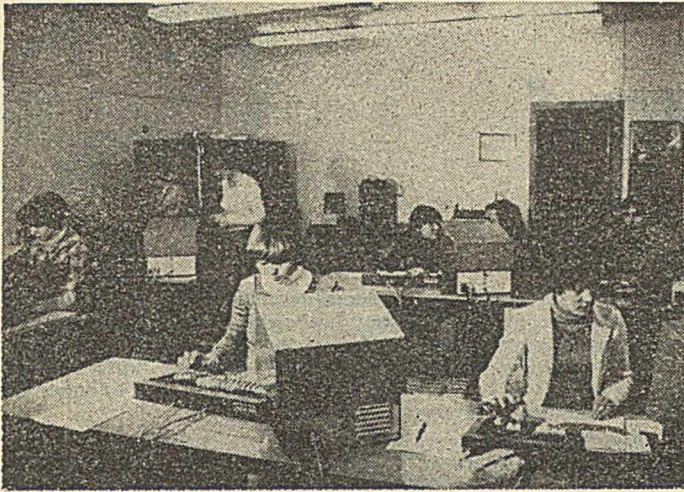
- system gospodarki materiałowej SARGM-77 — w 130 przedsiębiorstwach
- system gospodarki materiałowej dla przedsiębiorstw sprzętowo-transportowych SARGM — PST — w 11 przedsiębiorstwach



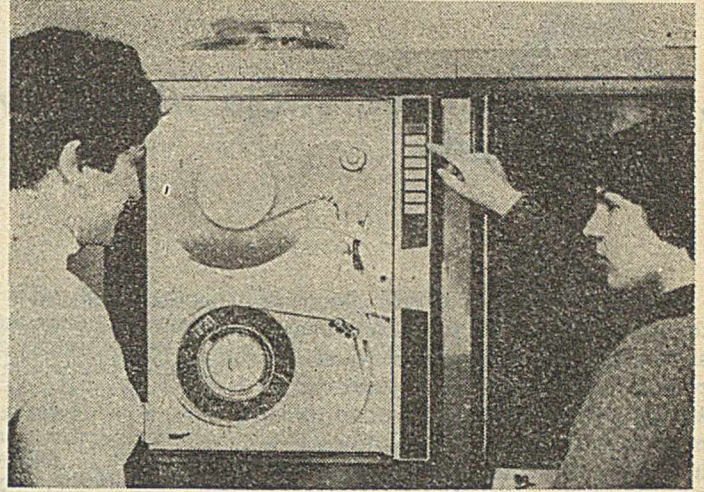
Największą troską dyrektora, dr. Zygmunta Klawittera, i wicedyrektora, mgr. inż. Tadeusza Nitki, jest aby tematy prac projektowych były ustalane przez zjednoczenia podległe CZBR



Dr inż. Alfred Lange (z lewej), projektant i działacz TNOiK, oraz mgr Stanisław Korendo zostali właśnie powołani na stanowiska kierowników działów



Parę lat temu takie urządzenie stanowiło ewenement w polskiej informatyce. Dzisiaj RC 3600 jest po prostu niezbędny dla BETOBRU, który eksploatuje wiele systemów ewidencyjnych na rzecz licznych przedsiębiorstw nie posiadających stacji przygotowania danych



Jednostka centralna systemu RC 3600 jest wysoce niezawodna (przy urządzeniu od lewej: Anna Szenejko i Mariola Tarnowska). W sytuacji, gdy BETOBR pozbył się nieomal zupełnie SOEMTRONÓW, awaria RC w okresie spiętrzenia terminowych prac byłaby katastrofą. Zdjęcia: Henryk KLEINZELLER

- system automatycznego rozliczania przedmiotów nietrwających — w 28 przedsiębiorstwach
- system ewidencji osobowej — w 61 przedsiębiorstwach
- system limitowania materiałów — w 181 przedsiębiorstwach
- system AWIZO-MOC, który nadzorował realizację 5,5 tys. inwestycji
- system PERT, którym nadzorowano wykonawstwo 32 zadań inwestycyjnych.

Oprócz eksploatacji systemów na rzecz gdańskiego ZBR, obsługiwano w podobnym zakresie również przedsiębiorstwa bydgoskie. W ramach wolnych mocy komputerów wykonywano prace na rzecz różnych instytucji województwa gdańskiego, a wśród nich dla Stoczni Remontowej w Gdańsku, Fabryki Gazomierzy w Tczewie i Fabryki Przekładni Samochodowych w Tczewie. Udzielano też wydatnej po-

mocy organizacyjnej — np. przy uruchamianiu stacji przygotowania danych dla zjednoczeń: opolskiego i bydgoskiego, a także prowadzono systematyczne szkolenie informatyków z branży. Sporo uwagi poświęcono też szkoleniu przyszłych użytkowników systemów informatycznych, a więc kadry kierowniczej przedsiębiorstw.

Przewiduje się, że w najbliższym czasie zostaną dokonane następujące zmiany w wyposażeniu technicznym: — pamięć operacyjna ODRY 1305 zostanie zwiększona do 128 K — pamięć zewnętrzna powiększy się o 8 jednostek dyskowych po 7,25 MB — urządzenia wyjścia zostaną wzmocnione dodatkową drukarką DW 325 — zainstaluje się 4 terminale ekranowe MERA 7950.

Niezbędne staną się inne uzupełnienia sprzętu, gdy BETOBR będzie kandydował na ośrodek resortowy. Chodzi tu przede wszystkim o:

- urządzenie do wprowadzenia danych na taśmę magnetyczną MERA 9150
- jednostkę dyskową dla systemu RC 3600

- 3 jednostki dyskowe po 30 MB
- komputer R-32 w konfiguracji gwarantującej teletransmisję.

Są to warunki warte spełnienia. Branżowy ośrodek eto jest przecież największy i najbardziej aktywny w informatyce resortu, skupia liczną, doświadczoną kadre, znającą się na problematyce rolnictwa. Dysponuje sporym potencjałem technicznym. Alternatywą jest tworzenie centralnego ośrodka od podstaw. Pochłonie to sporo czasu i jeszcze więcej pieniędzy. A z powołaniem centralnego ośrodka czującego nad informatyzacją rolnictwa nie można dłużej czekać.

Tekst: Krystyn BERNATOWICZ

„Zastosowanie komputerów w przemyśle”

W dniach 20—21 listopada 1978 r. odbyła się w Szczecinie konferencja naukowo-techniczna pod hasłem „Zastosowanie komputerów w przemyśle”, zorganizowana przez Szczeciński Oddział Polskiego Towarzystwa Cybernetycznego i Komitet Naukowo-Techniczny ds. Informatyki ROW NOT w Szczecinie.

Obrazy prowadzono w następujących czterech sekcjach problemowych: 1) Systemy czasu rzeczywistego, 2) Skomputeryzowane systemy zarządzania, 3) Projektowanie wspomagane komputerowo, 4) Modele, optymalizacja, symulacja komputerowa.

W zakończeniu obrad sformułowano następującą ocenę aktualnego stanu, kluczowych uwarunkowań oraz kierunków rozwoju zastosowań w przemyśle polskim:

1. Bardzo małe są aktualne możliwości zunifikowanych systemów minikomputerowych do zastosowań przemysłowych.

Ze względu na rosnące potrzeby gospodarki narodowej niezbędne jest bezwzględne podjęcie produkcji tego typu systemów modularnych z wyposażeniem umożliwiającym bezpośrednią współpracę z obiektami. W programie produkcyjnym należy uwzględnić mikroprocesory. Parametry minikomputera powinny być zbliżone do K 202 lub MERA 400. Moduły systemu powinny być produkowane masowo, aby umożliwić łatwe kompletowanie potrzebnych konfiguracji.

2. W dziedzinie przemysłowych zastosowań komputerów występuje niepokojące zjawisko wielokrotnego opracowywania tych samych

tematów. Niezbędne jest udoskonalenie wymiany informacji o pracach projektowych i wdrożeniowych realizowanych w różnych ośrodkach. Również Zjednoczenie MERA powinno szerzej informować o produkowanych urządzeniach i zamierzeniach produkcyjnych. Konieczne jest zintensyfikowanie działalności przedsiębiorstw kompletacji i dostaw systemów komputerowych w odniesieniu do zastosowań przemysłowych. Roli tej nie spełniają w dostatecznym stopniu istniejące przedsiębiorstwa Zjednoczenia MERA. 3. Szkolenie inżynierów, zwłaszcza projektantów, w zakresie informatyki jest niewystarczające i przebiega niewłaściwie. W programach szkolenia należy wyeksponować przygotowanie inżyniera nieinformatyka do korzystania z komputerów, zwłaszcza pod kątem umiejętności formułowania problemów inżynierskich oraz metod ich rozwiązywania.

Zajęcia praktyczne powinny odbywać się na nowoczesnym sprzęcie, uwzględniającym możliwość konwersacji, w tym również w formie graficznej. W szerszym zakresie należy wprowadzać metody symulacji komputerowej do badania problemów technicznych. Wyspecjalizowane ośrodki (szkoły wyższe, jednostki zaplecza naukowo-badawczego przemysłu) powinny prowadzić szkolenie dotyczące zastosowań bardziej zaawansowanych.

4. Uznano za bardzo wskazane organizowanie cyklicznych konferencji poświęconych przemysłowym zastosowaniom systemów komputerowych; postulowano, by organizatorem kolejnej konferencji był ośrodek szczeciński (JS)

„Należy tworzyć swoje własne systemy...”

W brytyjskim czasopiśmie „International Management” (nr 10/77) ukazał się artykuł zawierający wypowiedzi różnych specjalistów na temat wdrażania i eksploatacji systemów informatycznych. Ponieważ zagadnienia te interesują również naszych Czytelników, publikujemy poniżej wybrane fragmenty wspomnianego artykułu na podstawie opracowania Z. Moroza, opublikowanego w „Organizacji i Kierownictwie — przegląd czasopism zagranicznych”, z lutego 1978 (wszystkie podkr. — red.).

Wiele przedsiębiorstw uważa, że ich systemy komputerowe są zbyt skomplikowane. Nie jest przy tym ważne, czy są to systemy planowania, sterowania produkcją, czy informacyjne. Te same problemy pojawiają się wszędzie: w produkcji, finansach, handlu, transporcie czy dystrybucji towarów. Przedsiębiorstwa donoszą o niepowodzeniach we wdrażaniu i eksploatacji systemów. Nad kosztowne i skomplikowane instalacje komputerowe pracownicy często przekładają swoje własne systemy ręczne.

Kierownicy i operatorzy systemów narzekają na coraz bardziej skomplikowaną ich obsługę, są przytłoczeni masą papieru i wielką liczbą niepotrzebnych im danych. Eksperti twierdzą, że intensywne szkolenie pracowników niewiele pomoże. Przedsiębiorstwa powinny raczej pomyśleć o prostszych rozwiązaniach, zaprojektowanych z myślą o tych, którzy mają je stosować.

Niektórzy uważają, że pracowników trzeba nauczyć, jak należy tworzyć swoje własne systemy.

Peter Hermon z brytyjskich linii lotniczych (British Airways) zarzucił projektantom pogoń za techniką bez zwracania uwagi na cele, jakie zamierza się osiągnąć: *Wielu z nich postępuje tak jakby najważniejszym zadaniem było stosowanie najnowszych urządzeń komputerowych lub wdrażanie najnowszych programów, a przecież przede wszystkim chodzi o zaprojektowanie efektywnych i wydajnych systemów. Nowości techniczne należy stosować tylko wówczas, gdy jest to korzystne dla realizacji celów.*

Informatycy twierdzą, że urządzenia przemysłowe stają się coraz bardziej skomplikowane i aby temu sprostać potrzebne są systemy coraz bardziej złożone. Narzekają też, że wywierają na nich nacisk, aby opracowywali duże systemy, których wprowadzenie

ma usprawiedliwić wysoki koszt wyposażenia komputerowego. Często nawet jest wątpliwe, czy przedsiębiorstwo w ogóle potrzebuje takiego wyposażenia. Zdarza się, że systemy zaprojektowane na użytek komórek sztabowych przedsiębiorstw są instalowane we wszystkich działach operacyjnych przedsiębiorstw, aby rozłożyć ponoszone koszty.

Przedsiębiorcom wydaje się, że zastosowanie dużych systemów do zmieniających się okoliczności jest zbyt kosztowne. Zamiast tego projektanci tworzą „uzupełnienia” do systemów, zwiększając tylko stopień ich komplikacji...

Do pracy nad systemami angażuje się czasem przyszłych użytkowników, organizuje komitety lub grupy projektowe, w których skład wchodzi użytkownicy i informatycy. Firma Urwick Dynamics stwierdza: *Nasze badania wykazały, że takie komitety w dużym stopniu zawodzą. Nawet wtedy, gdy użytkownicy są mianowani przewodniczącymi tych komitetów, informatycy dążą do dominacji. Komitety takie zabierają tylko dużo czasu.*

Istnieje opinia, że problem rozwiążą minikomputery instalowane w pobliżu miejsca pracy użytkowników. Nie wszyscy jednak podzielają ten pogląd. *Kierownik, który sądzi, że łatwiej jest zrozumieć działanie sieci 20 minikomputerów niż jednego systemu, jest w błędzie. Problem jest dokładnie taki sam — mówi ekspert firmy Booz Allen*

Peter Hermon: *Musimy zrezygnować z wydajności systemów jako głównego celu. Powinniśmy zdecydować się na stosowanie prostszych programów i dokumentów. Teoretycznie korzystanie z urządzeń w sposób mało wydajny to marnowanie ich, ale w praktyce zmniejsza się wydajność, dzięki uniknięciu kosztownych opóźnień we wdrażaniu skomplikowanych systemów oraz dużych błędów, których przyczyną są takie systemy.*

Specjalny raport pt „Oddziaływanie systemów komputerowych na pracowników” stwierdza m.in., że specjaliści komputerowi powinni asystować kierownikom i pracownikom przedsiębiorstw przy projektowaniu ich własnych systemów, a nie odwrotnie. Takie rozwiązanie prowadzi do znacznego zmniejszenia trudności wdrażania i posługiwania się systemem.

Walery E. Daszko z radzieckiego instytutu cybernetycznego: *Wierzę, że w najbliższej przyszłości przedsiębiorstwa będą musiały zdecentralizować swoje ośrodki komputerowe i podporządkować komputery oraz odpowied-*

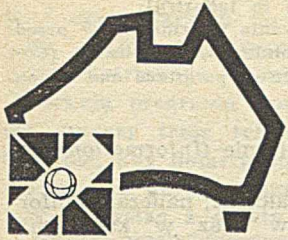
nich specjalistów kierownikom innych działów funkcyjnych. W ten sposób specjaliści ci będą asystować kierownikom w projektowaniu systemów i dostarczać im niezbędnych ekspertyz technicznych.

Projektanci systemów powinni bardziej uwzględnić wymagania biznesu, a ich projekty powinny ułatwiać realizację celów przedsiębiorstwa. Jest to przecież jedyny powód, dla którego instaluje się systemy — pisze jeden z autorów książki „The Effective Computer”.

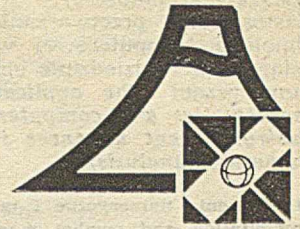
Projektanci systemów domagają się, aby projektowaniem systemów w większym stopniu interesowała się kadra kierownicza. *Kierownicy nie chcą i żadnego wysiłku, aby zrozumieć działanie systemu, a nawet nie chcą się o nich niczego nauczyć. Polegają na specjalistach i im zostawiają wdrażanie systemów. Żaden dobry kierownik produkcji nie kupi maszyny za 2 miliony dolarów, jeżeli nie wie, jak ona pracuje. A tak właśnie postępują kierownicy, którzy zgadzają się na instalację systemu komputerowego — ekspert firmy Booz Allen.*

Dzisiejsi menadżerowie muszą znaleźć wspólny język z projektantami i obsługą techniczną — mówi ekspert Philipsa. Oczywiście nie wszyscy oni muszą być ekspertami, lecz muszą chcieć nauczyć się języka ekspertów i ich sposobu myślenia.

Kierownicy wyższego szczebla są także krytykowani za brak zainteresowania systemami. *Nie wierzę, że kierownicy wyższego szczebla rzeczywiście zdają sobie sprawę z tego, jaki wpływ mają komputery na ich przedsiębiorstwa — mówi Aleksander Butrimienko z radzieckiej Akademii Nauk, specjalizujący się w problematyce przekazywania informacji — a przecież właśnie systemy informacyjne decydują o tym, jak są kierowane przedsiębiorstwa. Nie jest przy tym ważne, jaka jest formalna struktura organizacyjna. Kierownicy wyższego szczebla muszą patrzeć na systemy w ten sposób, jak to czynią w odniesieniu do finansów, produkcji, marketingu i wszystkich innych funkcji w przedsiębiorstwie.*



IFIP 80



Reprezentując interesy informatyki 39 krajów członkowskich, Międzynarodowa Federacja Przetwarzania Informacji (IFIP) zorganizowała dotąd siedem kongresów — w Paryżu, Monachium, Nowym Yorku, Edynburgu, Lublanie, Sztokholmie i Toronto. Ósmy światowy kongres IFIP 80 odbędzie się po raz pierwszy w rejonie zachodniego Pacyfiku.

Kongres IFIP 80 rozpocznie się w Tokio (Japonia) 6 października i będzie kontynuowany w Melbourne (Australia) od 14 do 17 października 1980 r.

W Tokio obrady będą się odbywały w Centralnej Hali Toshi (*Toshi Central Hall*), natomiast w Melbourne — w pawilonach wystawowych. Jednocześnie czynne będą w obu miastach wystawy, prezentujące światowy rozwój sprzętu informatycznego i jego zastosowań (w Tokio — w centrum handlowym HARUMI, w Melbourne — w pawilonach wystawowych).

Hasłem przewodnim imprezy będzie tym razem „Ku powszechności zastosowań” („Challenges of a computer presence”).

TEMATYKA PROGRAMU KONFERENCJI (PROGRAM AREAS)

1. Teoretyczne podstawy informatyki (*Theoretical foundations of information processing*)
Formalizacja koncepcji, metod matematycznych oraz teorii informatycznych (*Formalization of concepts, mathematical methods and theories of information processing*)

- Teoria obliczeń, języków formalnych oraz teoria automatów (*Theory of computation, formal languages and automata theory*)
- Analiza i synteza programów (*Analysis and synthesis of programs*)
- Semantyka systemów i języków programowania (*Semantics of systems and programming languages*)
- Analiza i optymalizacja algorytmów (*Analysis and optimization of algorithms*)
- Formalne aspekty sztucznej inteligencji, struktur danych i metodologii programowania (*Formal aspects of artificial intelligence, data structures and programming methodology*)

2. Architektura i konstrukcja komputerów (*Computer architecture and hardware*)
Postęp w technologii i jego wpływ na projektowanie systemów komputerowych (*Advances in technology and their influence on computer system design*)

- Architektura systemów (*System architecture*)
- Procesory specjalizowane (*Special purpose processors*)
- Technologia wejścia/wyjścia (*Input/Output technology*)
- niezawodność i tolerancja błędów (*Reliability and fault tolerance*)
- Technologia układów i pamięci (*Circuit and memory technology*)
- Pamięci o wielkiej pojemności (*Large capacity memories*)
- Automatyzacja projektowania układów LSI (*LSI design automation*)
- Ocena architektury (*Architecture assessment*)

3. Oprogramowanie (*Software*)

Programy i procedury ułatwiające opracowanie, eksploatację i dalszy rozwój systemów oprogramowania (*Programs and procedures which facilitate the development, operation and evolution of software systems*)

- Systemy operacyjne (*Operating systems*)
- Języki i systemy programowania (*Programming languages and systems*)
- Metody i narzędzia rozwoju oprogramowania (*Development tools and disciplines*)

- Organizacja oprogramowania (*Software organization*)
- Wydajność i niezawodność oprogramowania (*Software performance and reliability*)
- Mikroprogramowanie i oprogramowanie układowe (*Microprogramming and firmware*)
- Oprogramowanie mikrokomputerów (*Software for microcomputers*)

4. Baza danych i systemy informacyjne (*Data base and information systems*)
Modele, techniki i metodologie projektowania i wdrażania systemów informacyjnych (*Models, techniques and methodologies for the design and implementation of information systems*)

- Sprecyzowanie potrzeb informacyjnych (*Information requirements specification*)
- Architektura systemu bazy danych (*Data base system architecture*)
- Projektowanie bazy danych (*Data base design*)
- Rozproszone bazy danych (*Distributed data bases*)
- Systemy bazy danych i systemy przesyłania danych (*Data base and data communication systems*)
- Systemy sterująco-decyzyjne (*Decision control systems*)
- Translacja bazy danych (*Data base translation*)
- Modele i języki danych (*Data models and data languages*)
- Wyszukiwanie informacji (*Information retrieval*)

5. Sieci komputerowe i łączność (*Computer networks and communications*)
Zastosowanie połączonych komputerów i ich wpływ na środki łączności (*The use of interconnected computers and the implications for communication facilities*)

- Przetwarzanie rozproszone — sprzęt i oprogramowanie (*Distributed processing, hardware and software*)
- Techniki łączności: technologia i doświadczenia (*Communications techniques: technology and experience*)
- Połączenia systemów otwartych: standardy i protokoły (*Open-systems connections: standards and protocols*)
- Użycie minikomputerów i mikroprocesorów w sieciach (*Utilization of minicomputers and microprocessors in networks*)
- Zastosowanie sieci i podział zasobów (*Applications of networks and resource sharing*)
- Sieci lokalne (*Local networks*)
- Poczta elektroniczna (*Electronic mail*)
- Sieci danych cyfrowych (*Digital data networks*)

6. Komputeryzacja w nauce i przemyśle (*Computing in science and industry*)
Zastosowanie przetwarzania informacji, matematyki i technologii komputerowej w nauce i przemyśle. Aktualne osiągnięcia w dziedzinie obliczeń matematycznych i analizy numerycznej (*The application of information processing, mathematics and computer technology in science and industry. Recent advances in mathematical computation and numerical analysis*).

- Obliczenia numeryczne i manipulowanie symbolami (*Numerical computation and symbol manipulation*)
- Modele skomputeryzowane (*Computerized models*)
- Automatyka przemysłowa i robotyka (*Industrial automation and robotics*)
- Rozpoznawanie i przetwarzanie obrazów (*Pattern recognition and picture processing*)
- Przetwarzanie języka naturalnego i tekstów (*Natural language and text processing*)
- Projektowanie i produkowanie wspomaganie komputerem (*Computer aided design and manufacturing*)
- Grafika komputerowa (*Computer graphics*)
- Przetwarzanie głosu (*Voice processing*)
- Sprzęt interfejsowy (*Interface hardware*)

7. Zastosowania w gospodarce i administracji państwowej (*Business and government applications*)

Obecne techniki, nowe systemy, kluczowe zastosowania (*Present techniques, new systems, key applications*)

- Komputeryzacja, projektowanie i kierowanie (*Computerization, design and management*)
- Niezawodność i bezpieczeństwo, rewizja księgową (*Reliability and security, auditing*)
- Kontakty z użytkownikami (*Interaction with users*)
- Tanie systemy komputerowe w zarządzaniu (*Low cost computer systems in management*)
- Zastosowania gospodarcze: produkcja, sterowanie, prognozowanie, kadry (*Business applications: production, control, forecasting, personnel*)
- Zastosowania w centralnej i terenowej administracji państwowej (*Central and local government applications*)
- Komputery i bankowość, elektroniczny transfer kapitałów (*Computers and banking, electronic fund transfer*)
- Dystrybucja informacji (*Diffusion of information*)
- Użycie banków danych (*Use of data banks*)

8. Implikacje socjalne i ekonomiczne (*Social and economic implications*)

Badania przyczynkowe w przemyśle, administracji publicznej i organizacjach międzynarodowych (*Case studies in industry, public administration and international bodies*)

- Zmiany w strukturze socjalnej wskutek informatyki (*Changes in the social structure due to computing*)
- Informatyka i zatrudnienie (*Computing and employment*)
- Informatyka dla rozwoju ekonomiki (*Computing for developing economies*)
- Implikacje ekonomiczne i organizacyjne działalności legislacyjnej w zakresie praw osobistych jednostki i zabezpieczenia danych (*Economic and organizational implications of privacy and data security legislation*)
- Pomiar i porównanie wydajności i kosztów komputerów (*Measuring and comparing computer performances and costs*)
- Wzajemne oddziaływanie lokalnego i międzynarodowego przemysłu komputerowego (*Interaction between local and world-wide computer industries*)

UWAGI DO AUTORÓW REFERATÓW

Organizatorzy Kongresu zapraszają do nadsyłania referatów dotyczących wyżej podanych obszarów tematycznych.

Informacje powinny ściśle odnosić się do zagadnień projektowania lub użytkowania systemów komputerowych. Mogą one dotyczyć osiągnięć teoretycznych, nowych metod lub doświadczeń praktycznych.

Wszystkie nadesłane referaty będą analizowane z punktu widzenia ważności zagadnienia, ich oryginalności oraz przejrzystości prezentacji.

Aby pomóc autorom przygotowano „Instrukcje i wskazówki dla autorów”, które zainteresowani mogą otrzymać pod adresem:

Program Committee
IFIP Foundation
40, Paulus Potterstraat
1071 DB Amsterdam
THE NETHERLANDS

Przed wysłaniem referatów autorzy powinni uzyskać zgodę na ich wygłoszenie lub publikację.

● Informatyka, robotyka oraz zmiana wzorców w przemyśle (*Computing, robotics and changing patterns in industry*)

● Analiza kosztów oraz korzyści stosowania informatyki w przedsiębiorstwach prywatnych oraz w administracji publicznej (*Cost benefit analysis of computing in private corporations and in public administration*)

9. Przetwarzanie informacji i kształcenie (*Information processing and education*)

Ogólny wpływ komputerów na kształcenie; nauczanie informatyki studentów, profesjonalistów oraz w kształceniu ciągłym (*Influence of computers on education in general; teaching in formation processing to students, professionals and in continuing education*)

- Skomputeryzowany system nauczania (*Computerized education systems*)
- Nauczanie wspomaganie komputerem (*Computer aided instruction*)
- Wpływ metodologiczny na kształcenie (*Methodological impact on education*)
- Komputery w szkołach (*Computers in schools*)
- Nauczanie przetwarzania informacji: potrzeby użytkowników, przemysłu i nauki (*Teaching information processing: the needs of users, industry and science*)
- Rola terminologii i standardów w nauczaniu (*The role of terminology and standards in teaching*)
- Równowaga pomiędzy aspektami teoretycznymi i praktycznymi (*The balance between theoretical and practical aspects*)

10. Komputery w życiu codziennym (*Computers in everyday life*)

Trendy w obecności komputerów: w przedmiotach użytku osobistego, w urządzeniach domowych, a także jako elementy form sztuki oraz przedmiot powszechnie dostępny (*Trends of a computer presence: in personal possessions, in domestic appliances, as part of art forms, and as used by everyone*)

- Komputery i czas wolny (*Computers and leisure*)
- Komputery w domu (*Computers in the home*)
- Środki konwersacyjne i personifikowane (*Interactive and personalized media*)
- Sztuki wizualne i sceniczne (*The visual and performing arts*)
- Inteligentne telefony (*Intelligent telephones*)
- Uczestnictwo obywatelskie (*Citizen participation*)
- Osobiści asystenci (*Personal assistants*)
- Oddziaływanie psychologiczne (*Psychological impacts*)

Komitet programowy kongresu:

F. H. Summer — przewodniczący, A. P. Jersow — wice-przewodniczący, W. M. Turski — poprzedni przewodniczący, C. J. Pereira de Lucena, E. Goto, D. C. Tschritzis, G. N. Lance, H. E. Andersis, C. Berthet, M. G. Losano, P. Deussen, N. Negroponte, S. H. Lavington — redaktor materiałów konferencyjnych.

Program kongresu przewiduje trzy rodzaje sesji: pierwsza obejmuje referaty zamówione (ang. *invited papers*), odnoszące się do ogólnych problemów informatyki; dwie lub trzy sesje zostaną poświęcone referatom nadesłanym (ang. *submitted papers*), relacjonującym istotne aktualne osiągnięcia informatyki; ponadto odbędą się dyskusje panelowe (ang. *panel discussions*), analizujące obecny stan i główne trendy rozwoju informatyki, zakładając aktywny udział słuchaczy spotkania.

Niektóre z nadesłanych referatów zostaną przedstawione w obu miejscach obrad kongresu. Dlatego też Komitet Programowy Kongresu prosi autorów o podanie, czy pragną oni zaprezentować swe referaty w obu miejscach, dowolnym miejscu lub tylko w wybranym miejscu.

Referaty muszą być napisane i wygłoszone po angielsku.

Należy dostarczyć wyraźny maszynopis, pisany jednostronnie, z podwójnym odstępem.

Kompletny referat należy przesłać w 4 egzemplarzach do przewodniczącego sekcji tematycznej (Technical Area chairman), odpowiedzialnego za tę sekcję programu konferencji, której dotyczy treść referatu, w takim terminie, aby dotarły one do adresata nie później niż 1 grudnia 1979 r. Należy korzystać z poczty lotniczej.

Adresy przewodniczących sekcji podajemy obok.

Każdy egzemplarz referatu, dokładnie spięty zszywaczem w lewym górnym narożniku, powinien zawierać:

1. Stronę tytułową, zawierającą:

- tytuł referatu
- nazwisko, kraj, przynależność organizacyjną oraz dokładny adres pocztowy autora(ów)
- nazwę sekcji odpowiadającą tematycznie treści referatu (można wybrać tylko jedną z dziesięciu sekcji tematycznych)
- własnoręcznie podpisane następujące oświadczenie:

„Niniejszy referat — ani też jakakolwiek zbliżona do niego wersja — nie był i nie będzie nigdzie zgłoszony do publikacji, a w przypadku jego akceptacji zostanie wygłoszony w co najmniej jednym miejscu obrad ósmego Kongresu IFIP osobiście przez autora lub jednego ze współautorów”.

e) życzenie odnośnie do miejsca wygłoszenia referatu ze wskazaniem tylko jednej z następujących możliwości:

- prezentacja w Tokio lub Melbourne albo w obu miastach
- prezentacja w Tokio lub Melbourne lecz nie w obu miastach
- prezentacja tylko w Tokio
- prezentacja tylko w Melbourne

2. Strona zawierająca streszczenie w języku angielskim o objętości do 100 słów

3. Tekst ostatecznej wersji referatu, napisany w języku angielskim na maszynie

4. Ilustracje z objaśnieniami, ponumerowane kolejno i odpowiednio zaznaczone w tekście (uwaga: nie przysyłać oryginalnych fotografii lub grafiki artystycznej przed ostatecznym zakwalifikowaniem referatu).

W górnym lewym narożniku każdej strony należy zamieścić nazwisko autora, a w prawym górnym narożniku — kolejny numer strony (streszczenie powinno być na stronie 2).

Referat łącznie ze wszystkimi ilustracjami, wzorami i odsyłaczami nie może przekroczyć 4000 słów. Jedna ilustracja odpowiada 250 słowom tekstu.

Informacja o akceptacji lub odrzuceniu zostanie wysłana do autorów do 15 marca 1980 r. (w przypadku referatów opracowanych przez kilku autorów cała korespondencja będzie adresowana do autora wymienionego na pierwszym miejscu), a wkrótce potem będzie opublikowany wstępny program kongresu. Ze względu na znaczną liczbę spodziewanych referatów wyklucza się możliwość zwrotu materiałów.

W niedługim czasie po akceptacji referatów autorzy będą zobowiązani do powtórnego przepisania na maszynie swych referatów w formie przystosowanej do fotoreprodukcji w materiałach kongresu. Szczegółowa instrukcja łącznie z matrycami maszynowymi zostanie autorom przesłana pocztą.

Autorzy będą zobowiązani do dostarczenia przepisanych referatów oraz niezbędnych oryginałów fotografii, grafiki itp. w ciągu dwóch tygodni od otrzymania przesyłki z instrukcją.

Komitet Programowy zastrzega sobie prawo dokonania niewielkich zmian wydawniczych albo ponownego zwrócenia się do autorów z prośbą o zredagowanie swych referatów w sposób zgodny z wymaganiami wydawniczymi.

Wysyłając referat, należy na kopercie napisać:
PROGRAM COMMITTEE, 8th WORLD COMPUTER CONGRESS, adresując do przewodniczącego wybranej sekcji tematycznej.

Przed wysyłką należy jeszcze raz sprawdzić, czy referat skierowany jest do przewodniczącego właściwej sekcji. Błędne skierowanie może spowodować przekroczenie ustalonego terminu składania, a tym samym odrzucenie referatu.

W przypadku jakichkolwiek wątpliwości co do prawidłowego wyboru sekcji tematycznej, należy wysłać referat odpowiednio wcześniej — tak, aby dotarł do przewodniczącego sekcji ok. 15 listopada br. Pozwoli to nie przekroczyć ostatecznego terminu przyjmowania referatów (1 grudnia br.).

Adresy przewodniczących poszczególnych sekcji tematycznych:

Sekcja 1

Prof. C. J. P. de LUCENA
Departamento de Informatica
Pontificia Universidade Catolica
Rua Marques de S. Vincente, 225
GAVEA-CEP 22453
Rio de Janeiro, RJ
Brazil

Sekcja 2

Prof. E. GOTO
Department of Information Science
University of Tokyo
7-3-1 Bunkyo-ku
TOKYO 113
Japan

Sekcja 3

Prof. A. P. ERSHOV,
Computing Center
NOVOSIBIRSK 630090
U.S.S.R.

Sekcja 4

Prof. D. C. TSICHRITZIS
Dept. of Computer Science
University of Toronto,
TORONTO, Ontario,
Canada M5S 1A7

Sekcja 5

Dr. G. N. LANCE,
CSIRO
Division of Land Use Research
P. O. Box 1666
CANBERRA CITY, 2601
Australia

Sekcja 6

Dr. H. E. ANDERSIN
Valmet Oy
Punanotkonkatu 2
HELSINKI
Finland

Sekcja 7

Prof. C. BERTHET
Computing Center
University of Paris — Dauphine
Place du Maréchal de Lattre de Tassigny
75775 PARIS CEDEX 18
France

Sekcja 8

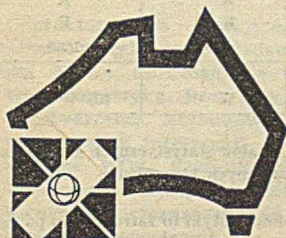
Prof. M. G. LOSANO
State University of Milan
c/o Siemens Data
Viale Monza 347
I-20128 MILAN
Italy

Sekcja 9

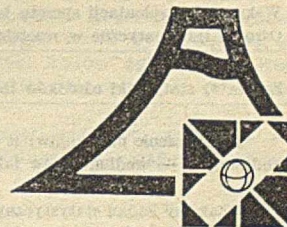
Prof. P. DEUSSEN,
Institut für Informatik I
Universität Karlsruhe
Postfach 6380
D-7500 KARLSRUHE 1
Federal Republic of Germany

Sekcja 10

Prof. N. NEGROPONTE
The Architecture Machine Group
M. I. T. ROOM 9-516
Cambridge, Mass., 02139,
U. S. A.



Zgodnie z uchwałą Komitetu Informatyki PAN, który reprezentuje Polskę w federacji IFIP, autorzy krajowi powinni przed przetłumaczeniem na język angielski przesłać teksty swych referatów do wstępnego zaopiniowania w terminie do 1 września br. na ręce doc. dr. hab. inż. J. Golińskiego, Zjednoczenie Informatyki, 02-078 Warszawa, ul. Krzywickiego 34.



Motto:

*Piekło jest wybrukowane
dobrymi... sprawozdaniami
(zasłyszane)*

Narodziny nowego działu statystyki gospodarczej

Powyższy tytuł można uważać za symboliczny. Oto bowiem z początkiem 1979 roku weszły w życie nowe wzory sprawozdawczości obowiązkowej ośrodków informatyki. Mniej więcej w tym czasie, kiedy niniejsza recenzja ukazuje się w druku, Ośrodek Badawczo-Rozwojowy Systemu Państwowej Informacji Statystycznej opublikuje piąty tom zestawień sumarycznych o krajowym parku komputerowym i działalności ośrodków informatycznych. Będzie to ostatnie „tabularium” sporządzane według starych wzorów.

Pierwsza obszerniejsza informacja OBR SPIS¹⁾ obejmowała 24 tablice ogólne dla 1974—75 w części opisowej oraz 16 tabulogramów za rok 1975 w części szczegółowej, ujętych w dwu przekrojach „brzegowych” — resortowym i terenowym, już według nowego podziału administracyjnego kraju. Umownie więc można przyjąć rok 1974 za datę

1) Informatyka i ośrodki informatyki 1975. Seria „Materiały Statystyczne”; Główny Urząd Statystyczny, Ośrodek Badawczo-Rozwojowy SPIS; Warszawa, czerwiec 1976; maszynopis powielany 73 str.; nakład 500 egz.

narodzin statystyki informatycznej w Polsce²⁾. Jest to o tyle ciekawe, że właściwie jesteśmy nadal jednym z nielicznych krajów, w których prowadzone są systematycznie głębsze opracowania statystyczne z działalności ośrodków komputerowych. Słynne „Statystyki Diebolda” odnoszą się tylko do liczby komputerów zakupionych, nie wnikając w faktyczny stopień wykorzystania; są to więc jakby działły statystyki handlowej, a nie metodologicznie odrębna statystyka informatyczna. Bardziej systematyczne ujęcie charakteryzuje zestawienia publikowane we Francji, ale już w roczniku statystycznym ONZ głucho o komputerach.

2) Właściwie jednak statystyka informatyczna rodziła się na przełomie lat 1969/70, kiedy to wprowadzono rozszerzone formularze sprawozdawcze z miesięcznej działalności ośrodków komputerowych — zanim sprawę ostatecznie przejął GUS. Gdyby jednak sprawę potraktować bardziej metodologicznie, to można dojść do wniosku, że statystyka informatyczna nadal się rodzi — choćby dlatego, że informatyka nadal ulega głębokim przeobrażeniom jakościowym, ba, wykazuje nawet wyraźne mankamenty deficyjne.

Porównywalność przekrojów brzegowych w materiałach statystycznych za lata 1975—1977

Charakterystyka merytoryczna tabulogramu	Numer tabulogramu			Zastosowane przekroje*) w latach:		
	1975	1976	1977	1975	1976	1977
Dane sumaryczne o ośrodkach Informatyki	—	—	1.	—	—	RW
Ośrodki wg klas zatrudnienia	1.	1.	2.	RW	FRW	FDRW
Ośrodki wg klas sprzętu	2.	4.	3.	RW	FRW	ZRW
Ośrodki wg klas zastosowań	—	2.	4.	—	R	ZR
Sprzęt komputerowy wg klas konstrukcyjnych	—	3.	—	—	R	—
Sprzęt komputerowy wg generacji	3.	5.	5.	RW	FRW	ZDR
Sprzęt komputerowy wg pochodzenia	3.	5.	5.	RW	FRW	ZDR
Sprzęt komputerowy wg wieku	5.	5.	6.	T	F	T
Sprzęt komputerowy wg klas pamięci operacyjnej	4.	6.	7.	R	FR	R
Sprzęt licząco-analityczny wg typów	6.	9.	8.	RW	FR	R
Sprzęt przygotowywania nośników danych	8.	10.	9.	RW	FR	ZR
Sprzęt teledacyjny wg klas urządzeń	7.	8.	10.	R	F	FZR
Rozliczenia czasu komputerowego wg składników	9.	11.	11.	TRK	FR	TFZR
Rozliczenie czasu minikomputerowego wg składników	—	12.	12.	—	PR	TFZR
Zatrudnienie wg specjalności	11.	13.	13.	DRW	FRW	FZDRW
Ruch zatrudnionych w ośrodkach Informatyki	12.	14.	14.	DRW	FR	FR
Place przeciętne wg grup specjalności	13.	15.	15.	R	FR	FZR
Wartość parku maszynowego wg grup sprzętu	10.	16.	16.	DRW	R	DRW
Wartość sprzedanych usług	—	17.	17.	—	R	FR
Koszty ruchu ośrodków wg składników	16.	19.	18.	DR	FR	FR
Nakłady rozwojowe wg grup kosztów	15.	20.	19.	DR	FR	FDRW
Tematyka zastosowań wg klas systemów	14.	—	—	DRW	—	—
Rozliczenie czasu zatrudnionych projektantów (programistów)	—	18.	20.	—	K	K
Wskaźniki eksploatacji sprzętu komputerowego	—	18.	20.	—	K	K
Ośrodki informatyczne w resorcie MNSZWIT	—	—	21.	—	—	różne
Zakres**) statystyki ośrodków Informatycznych	—	różne	—	0	W	(0) W

*) Oznaczenie przekrojów: R — według resortów, W — według województw w nowym podziale administracyjnym, D — według działów statystycznych gospodarki narodowej, T — według typów fabrycznych sprzętu, F — według form finansowania, Z — według klas zatrudnienia, K — według innych klasyfikacji

**) Zakresy badań statystycznych: O — ogółem wszystkie ośrodki informatyki, oprócz MON i MSW; W — tzw. większe ośrodki informatyki (o zatrudnieniu ponad 4 osoby) o znaczeniu bezpośrednio gospodarczym

Druga informacja OBR SPIS³⁾, zaskoczyła nieco innym ujęciem „główek” tabulogramów i zmianą klas statystycznych, co zresztą wynikało ze szlachetnego skąd inąd dążenia do uzyskania nowych przekrojów brzegowych, gdyż dotychczas zestawione szeregi statystyczne były zbyt krótkie. W sumie więc na lata 1975—77 przypadł okres poszukiwania adekwatnych ujęć tabelarycznych. Ponadto podkreślić należy fakt ograniczonego kolportażu rzeczonych opracowania — rozsyłanego głównie do resortów, władz wojewódzkich, różnych komisji itp. Odbiorcy odczuli jednak pewien niepokój, wynikający z niepełnej porównywalności obu opracowań.

Pozornie niepokój ten, a niekiedy wręcz niedosyt, wzrósł po ukazaniu się trzeciego opracowania⁴⁾. Zawiera ono 29 tablic oraz 21 tabulogramów, ale w jeszcze innych układach. Można by złośliwie określić tego rodzaju praktyki jako układanie krzyżówek. Recenzent jednak rozumie doskonale, że jest to jedynie wynik poszukiwań najdogodniejszych przekrojów statystycznych. Łatwo mówić, że np. statystyka rolna jeżeli zmienia układ opracowań, czyni to w sposób niemal niedostrzegalny — ale statystycy rolni dysponują blisko 100-letnim doświadczeniem swych poprzedników. Statystycy informatyki nie mieli żadnych poprzedników, a ich materiały porównywalne nie sięgają nawet 1/20 tego okresu! Na dobro Autorów ostatniego opracowania — a są nimi Ryszard Gujski, Romana Sachnowska i Halina Kroh — należy jednak zaliczyć odczuwalny fakt nawiązania do układu materiałów za rok 1975. Obrazowo ilustruje to załączona tablica porównawcza.

Tak się złożyło, że uczestnicząc w pracach analitycznych pewnego zespołu koordynacyjnego, recenzent miał możliwość nawiązać osobisty kontakt z Autorami i przekazać im bezpośrednio szereg uwag szczegółowych — spotykając się zresztą z pełnym zrozumieniem. W tym względzie recenzent oczekuje, że następna publikacja, omawiająca wyniki działalności informatycznej 1978 roku, ukaże się już w bardziej jednolitym układzie. Tutaj należy jednakże podkreślić pewien dosyć istotny moment psychologiczny.

Dla osób należących do krajowego aparatu statystycznego termin „jednolity” kojarzy się przede wszystkim z formą informacji sprawozdawczej. Jednakże dla odbiorców opracowań statystycznych termin „jednolitość” kojarzy się przede wszystkim z porównywalnością układów tabulogramów w różnych latach.

Zachęcam do lektury kolejnych opracowań OBR SPIS, bo być może w wyniku przetrwania zawartych w nich układów tabelarycznych, które po prostu należy traktować jako propozycje do dalszej dyskusji, informatycy dopracują się bardziej syntetycznych wskaźników. Dopiero wówczas będzie można naprawdę mówić o jednolitych szeregach statystycznych dla informatyki. Co nie zmienia faktu, że nie tylko niżej podpisany odczuwa potrzebę sporządzenia choćby prowizorycznych szeregów statystycznych dla działalności krajowych ośrodków informatycznych za lata 1976—1976—1977—1978, i to w dodatku nie tylko w przekrojach brzegowych.

Zachęcam do lektury kolejnych opracowań OBR SPIS, bo być może w wyniku przetrwania zawartych w nich układów tabelarycznych, które po prostu należy traktować jako propozycje do dalszej dyskusji, informatycy dopracują się bardziej syntetycznych wskaźników. Dopiero wówczas będzie można naprawdę mówić o jednolitych szeregach statystycznych dla informatyki. Co nie zmienia faktu, że nie tylko niżej podpisany odczuwa potrzebę sporządzenia choćby prowizorycznych szeregów statystycznych dla działalności krajowych ośrodków informatycznych za lata 1976—1976—1977—1978, i to w dodatku nie tylko w przekrojach brzegowych.

Adam B. EMPACHER

3) Informatyka i ośrodki informatyki 1976. Seria „Materiały Statystyczne”; GUS, OBR, SPIS; Warszawa, czerwiec 1977; maszynopis powielany 54 str.; nakład 500 egz.

4) Informatyka i ośrodki informatyki 1977. Seria „Materiały Statystyczne”; GUS, OBR, SPIS; Warszawa, czerwiec 1978; maszynopis powielany 72 str.; nakład 300 egz.

Bibliografia wydawnictw polskich z dziedziny informatyki

● Informatyczny system sterowania zaopatrzeniem produkcji w narzędzia — KONOWROCKI T. Wyd. Instytutu Organizacji Przemysłu Maszynowego, Warszawa 1977 r., s. 70

Model systemu sterowania zaopatrzeniem produkcji w narzędzia. Modele zapasów stosowane w algorytmie sterowania zaopatrzeniem produkcji w narzędzia. Badania przydatności modelu sterowania zaopatrzeniem produkcji w narzędzia na danych rzeczywistych w przedsiębiorstwie. Informatyczny system sterowania zaopatrzeniem produkcji w narzędzia.

Opracowanie to jest fragmentem pracy doktorskiej wykonanej przez autora w Szkole Głównej Planowania i Statystyki pod kierunkiem prof. dr. inż. Z. Zbichorskiego. Praca przeznaczona jest dla projektantów systemów informatycznych zarządzania.

● Skomputeryzowany system informacji o jakości węgla w pokładach — NOWAK Z. i inni. Wyd. Głównego Instytutu Górniczego, Katowice 1977 r., s. 15, cena 23 zł. Praca Głównego Instytutu Górniczego. Komunikat nr 683

Charakterystyka istniejącego systemu gromadzenia informacji o jakości węgla w pokładach. Założenia metodologiczne do budowy banku informacji o jakości węgla w pokładach oraz jego charakterystyka. Programy banku jakości węgla i przykłady zadań przetwarzania. Rozbudowa i aktualizacja zbioru danych. Przykłady opracowania informacji o jakości węgla za pomocą systemu programów BJW. Kierunki dalszych prac nad rozbudową systemu. Zastosowano komputer ODRA 1204.

Materiały przeznaczone są dla projektantów systemów informatycznych zarządzania.

● Programowanie w języku ASSEMBLER Jednolitego Systemu elektronicznych maszyn cyfrowych. Cz. II Opis języka — ATANKO J. Wyd. Politechniki Wrocławskiej, Wrocław 1978 r., s. 216, cena 30 zł. Biblioteka WASC

Opis języka ASSEMBLER i translatora. Elementy sterowania programami przez supervisor. Przykład programu. Uwagi o systemie operacyjnym OS/JS. Dodatki: Możliwe definicje stałych. Lista rozkazów EMC R-32.

Materiały przeznaczone są dla programistów.

● Podstawy kodowania nadmiarowego — DRÓZDZ J. Wyd. Politechniki Warszawskiej, Warszawa 1978 r., s. 220, cena 14 zł

Przesyłanie danych. Kodowanie nadmiarowe. Kody liniowe. Dokodowanie detekcyjne kodu liniowego. Dokodowanie korekcyjne kodu liniowego. Przegląd podstawowych rodzajów kodów liniowych. Kod liniowy jako przestrzeń liniowa. Macierzowy zapis kodu liniowego. Graniczne właściwości kodów liniowych. Kody cykliczne. Macierzowy zapis kodu cyklicznego. Sposoby kodowania. Detekcyjne dokodowanie kodów cyklicznych. Realizacja techniczna koderów i dekoderów kodu cyklicznego. Kody cykliczne do korekcyj błędów. Algebraiczne właściwości kodów cyklicznych. Kody Bose-Chaudhuri — Hocguenghema (BCH). Algebraiczne dekodowanie kodów BCH.

Skrypt przeznaczony jest dla studentów starszych semestrów kierunku „Telekomunikacja”, w szczególności specjalizacji „teleinformatyka”.

Oprac. A.K.



Oprogramowanie komputerów JS

Dział Serwisu Oprogramowania ELWRO-SERWIS dysponuje aktualnie następującym oprogramowaniem podstawowym dla komputerów JS:

- system DOS/JS, wersja 1.3 mod 7
- zadania kontrolne w systemie DOS/JS
- system OS/JS, wersja 3.0
- zadania kontrolne w systemie OS/JS
- biblioteka testów, wersja 004
- testy lokalizacji uszkodzeń DTLU

Z nowych pozycji oprogramowania użytkowego dostępne są aktualnie:

- system wyszukiwania informacji naukowo-technicznej w systemie OS — WINT
- konwersja programów źródłowych COBOLU ODRA 1300 na programy źródłowe COBOLU JS w systemie OS
- konwersja programów źródłowych FORTRANU ODRA 1300 na programy źródłowe FORTRANU JS w systemie OS.

Urządzenia pomocnicze dla ośrodków obliczeniowych

W Centrum MERA-ELWRO został opracowany zestaw urządzeń do transportu i przechowywania maszynowych nośników informacji. Zestaw ten składa się z szaf i regałów o modułowej konstrukcji, pojemników dostosowanych do taśm magnetycznych, pakietów dyskowych i kart dziurkowanych oraz wózków służących do transportu tych nośników. Regały i szafy montowane są ze zunifikowanych elementów konstrukcyjnych, umożliwiających użytkownikom montowanie najrozmaitszych zestawów magazynowych. Montaż szaf i regałów przeprowadzany będzie na wskazanych u użytkownika pomieszczeniach docelowych.

Na życzenie klientów Centrum MERA-ELWRO udziela bliższych informacji na temat powyższego sprzętu. Celem zbilansowania potrzeb krajowych i ustalenia kolejności uruchomień poszczególnych wyrobów. Zamówienia na powyższe zestawy prosimy kierować pod adresem: Biuro Centralnych Dostaw MERA-ELWRO, ul. Ostrowskiego 32, 53-238 Wrocław.

Zmiany konstrukcyjne w MTS 304-2

W ostatnim czasie w MTS 304-2 zostały opracowane i obecnie są wprowadzane do produkcji zmiany mające na celu poprawę niezawodności systemu. Główną treścią tych zmian jest zabezpieczenie systemu przed wyłączeniem jednostki taśmowej przy wciśniętym klawiszu GOT. W dotychczasowym rozwiązaniu w tej sytuacji MTS „widząc” jednostkę taśmową jako gotową (linia NGOT = 0) przyjmował operację, lecz ich nie wykonywał blokując tym samym pracę pozostałych jednostek taśmowych. W nowej wersji MTS, w dalszym ciągu sygnalizuje systemowi gotowość wyłączonej jednostki taśmowej, lecz po przyjęciu operacji następuje przerwanie (ustawienie linii B). Dla operacji z przesuwem taśmy (z prędkością roboczą równocześnie z przerwaniem) ustawione zostają statusy ZAKOŃCZENIE oraz BŁĄD. System operacyjny reaguje „odrzucając” jednostki taśmowej wypisując na monitorze komunikat dla operatora (FALL TYPE 2). Praca na pozostałych jednostkach taśmowych odbywa się normalnie.

Wprowadzone zostało zerowanie przerytnika BŁ. STER poprzez odczyt STATUSU P1. Eliminuje to konieczność interwencji operatora (dotychczas BŁ. STER był zerowany tylko specjalnym klawiszem). Zwiększony został także czas, po którym zapalany jest status CZYSTA TAŚMA, w celu zabezpieczenia przed zapaleniem się tego statusu przy wielokrotnej próbie zapisu bloku na taśmie ze zniszczonym nośnikiem.

Umowa serwisowa MERA-ELWRO — MERA-ELZAB

Biuro Obsługi Technicznej ELWRO-SERWIS zawarło umowę serwisową z Zakładem Urządzeń Komputerowych MERA-ELZAB w Zabrzu, na mocy której ELWRO przejmuje serwis monitorów ekranowych MERA7900, wchodzących w skład zestawów komputerowych rozprawdzanych przez elwrowskie Biuro Generalnych Dostaw, a szkolenie użytkowników prowadzi Ośrodek Szkoleniowy ELWRO-SERWIS.

Kto może być przyjęty na kursy?

Ośrodek Szkoleniowy ELWRO-SERWIS informuje, iż kandydaci ubiegający się o przyjęcie na kursy z zakresu obsługi technicznej maszyn cyfrowych powinni mieć ukończone studia wyższe o specjalności: maszyny matematyczne, automatyka lub elektronika przemysłowa.

Przed przyjęciem na kurs kandydaci zostaną poddani egzaminowi testowemu, obejmującemu następujące zagadnienia:

- elementy półprzewodnikowe dyskretne (diody, tranzystory, tyrystory)
- cyfrowe układy scalone TTL (elementy NAND, NOR, przerzutniki typu D i J-K)
- zasilanie (układy prostownicze, układy filtrujące napięcie, stabilizatory)
- elementy arytmetyki maszyn cyfrowych (systemy dwójkowe, ósemkowe, szesnastkowe, sposoby przedstawiania liczb ujemnych).

Kandydaci, którzy nie mają odpowiedniego wykształcenia lub nie zaliczą testu wstępnego, nie będą przyjęci.

Działalność Rady Kompleksowej Obsługi JS EMC

W październiku 1978 roku odbyły się posiedzenia poszczególnych sekcji Rady Kompleksowej Obsługi JS EMC. I tak: w Pradze, obradowała Tymczasowa Grupa Robocza ds. części zamiennych, w Ahlbecku (NRD) — sekcja ds. szkolenia, w Leningradzie — sekcja ds. obsługi technicznej. Podsumowanie obrad tych sekcji było tematem czwartej sesji Rady Kompleksowej Obsługi, która obradowała w listopadzie w Tallinie.

Ponadto w czwartym kwartale 1978 r. odbyły się posiedzenia dwustronne w ramach współpracy Polski z innymi krajowymi organizacjami Obsługi Technicznej JS EMC: w Charkowie i Wrocławiu, w Budapeszcie, w Lipsku, w Pradze.

Na spotkaniach przyjęty został plan dwustronnej współpracy na rok 1979.

Wszystkie materiały opracował Jerzy JANKOWSKI

MAREK LASOTA

Ośrodek Badawczo-Rozwojowy Systemu
Państwowej Informacji Statystycznej przy GUS
Warszawa

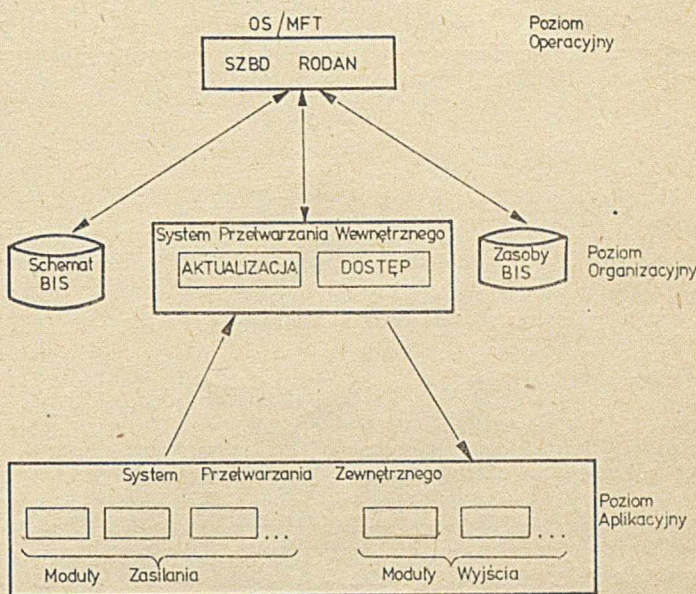
Bank danych statystycznych w technologii DBTG CODASYL

Specyficzne własności systemów przetwarzania danych statystycznych (duża objętość zbiorów danych, proste przetwarzanie, stabilna forma i regularny spływ danych, nieprecyzyjny formalny model danych) sprawiają, że rozwój technologii baz danych postępuje w tej dziedzinie stosunkowo powoli. Nieustanna poprawa parametrów sprzętu komputerowego, coraz większe potrzeby użytkowników, a także coraz większe trudności z przygotowaniem oprogramowania o odpowiedniej jakości i w odpowiedniej ilości klasycznymi metodami wpływają jednak na zmianę stanu rzeczy. Jednym ze sposobów wyjścia na przeciw tym problemom jest szersze wykorzystanie możliwości, jakie oferuje dostępne na rynku uniwersalne oprogramowanie systemów zarządzania bazą danych (SZBD), implementujące w większości przypadków sieciową metodę DBTG CODASYL [1], [2].

W ramach prac nad Rządowym Systemem Państwowej Informacji Statystycznej podjęto badania zmierzające do praktycznego określenia przydatności sieciowego SZBD w dziedzinie przetwarzania danych statystycznych. Pierwszym krokiem w tym kierunku było opracowanie doświadczalnej bazy danych statystycznych BASTA, obejmującej dane rocznej, ogólnokrajowej sprawozdawczości przedsiębiorstw budowlano-montażowych. Wykorzystano do tego celu system DMS 1100, zainstalowany na maszynie UNIVAC 1106. Pozytywne rezultaty tego eksperymentu [3] zachęciły do kontynuowania prac w tym kierunku. Ich efektem stało się utworzenie Banku Informacji Statystycznej (BIS) w oparciu o komputery Jednolitego Systemu oraz SZBD RODAN, zrealizowany w Ośrodku Badawczo-Rozwojowym Informatyki. Zebrane przy tej okazji doświadczenia, związane ze specyficznymi problemami budowy sieciowego banku danych statystycznych, stanowią treść tego artykułu.

ARCHITEKTURA BANKU DANYCH BIS

Zestaw elementów tworzących bank danych BIS i ich wzajemne powiązania (rys. 1), można podzielić na trzy poziomy. Najwyższy z nich — operacyjny — zajmują: system operacyjny OS MFT i SZBD RODAN, tworząc środowisko i dostarczając narzędzi do produkcji oprogramowania użytkowego. Poziom środkowy — organizacyjny — to zbiory danych i oprogramowanie decydujące łącznie o tożsamości banku. Znajdujemy tu więc bazy danych (jej opis i zasoby) oraz system przetwarzania wewnętrznego — uniwersalny pakiet programów, realizujący funkcje wprowadzania, aktualizacji i wyszukiwania danych. Warto dodać, że tylko oprogramowanie tego poziomu komunikuje się bezpośrednio z bazą danych. Tu zatem kończy się oddziaływanie mechanizmów (głównie metod dostępu do danych) właściwych dla SZBD RODAN. Najniższy poziom — aplikacyjny — jest zestawem oprogramowania użytkowego, zapewniającego zasilanie banku danych i tworzenie dokumentów wyjściowych. Na tym poziomie wykonuje się więc czynności, typowe dla statystycznego systemu przetwarzania, takie jak sprawdzanie, konwersja i tabulowanie danych.



Rys. 1. Architektura banku danych BIS

Jak widać ogólna koncepcja banku danych BIS nie jest ściśle związana z systemem przetwarzania danych statystycznych. Ukierunkowana jest natomiast na ogólne problemy dostępu do dużego zbioru danych. Cecha ta decyduje o możliwości zastosowania BIS w innych niż statystyka dziedzinach zastosowań, na przykład dla systemów informacyjnych resortów.

Podstawowe cechy bazy danych, wchodzącej w skład banku danych BIS, można określić następująco:

- możliwość sukcesywnej rozbudowy podczas 5—6 okresów sprawozdawczych
- duża liczba jednostek badania (rzędu 10000)
- duża liczba danych dotyczących jednej jednostki (do 2000 na jeden okres sprawozdawczy)
- różnorodny sposób przetwarzania, związany z zaspokajaniem potrzeb różnych użytkowników.

Istotnym problemem w aspekcie tych wymagań jest sposób organizacji zasobów bazy danych, niosących różne informacje i wykorzystywanych w różny sposób. Szczególnie ważną sprawą jest określenie postaci głównych zbiorów danych. Zarówno rozważania teoretyczne, jak i doświadczenia z bazą danych BASTA potwierdziły celowość nadania im nieskomplikowanej struktury wewnętrznej. Z drugiej strony, dla zapewnienia bazy danych odpowiednich własności eksploatacyjnych niezbędne jest wprowadzenie możliwości efektywnej selekcji danych według rozmaitych kryteriów. Metodą pogodzenia tych przeciwnych wymagań stało się utworzenie bazy danych BIS z dwóch rodzajów komponentów: masywów danych oraz wiążącego je masywu katalogowego (rys. 2).

Punktem odniesienia dla każdej z tych czynności jest stan bazy danych z roku poprzedniego. Wszelkie zmiany i uzupełnienia mogą polegać jedynie na dopisywaniu nowych informacji (technologicznie zrealizowano to głównie przez dopuszczenie możliwości duplikowania pewnych rekordów). Zapewniona jest przy tym możliwość określenia, jakie zmiany zasły w bazie danych w dowolnym czasie. Dzięki temu, oprócz dobrych własności eksploatacyjnych, zapewniono bazie danych pełną ochronę integralności.

Drugim modulem systemu przetwarzania wewnętrznego jest **moduł dostępu**. Zadaniem jego jest wybór wskazanych danych z bazy danych według kryteriów obowiązujących jednostki badania. Przewiduje się tu trzy zasadnicze tryby przetwarzania:

- 1) katalogowy — wybór jednostek badania jest określony na podstawie ich cech klasyfikacyjnych, z których większość uwzględniono w katalogu grupowań
- 2) rejestrowy — wybór jednostek badania następuje na podstawie zbioru identyfikatorów, podanego przez użytkownika
- 3) wartościowy — wybór jednostek badania określony jest wartością danych lub relacją między nimi.

Dla każdego z tych trybów możliwe są różne warianty przetwarzania wieloletniego. Informacje zawarte w bazie danych umożliwiają m.in. wyprowadzanie danych w kolejnych latach według stanu faktycznego, według stanu roku bazowego, w układzie maksymalnie porównywalnym itd.

Zasadą działania modułu dostępu jest dwuetapowy tryb selekcji danych. W pierwszym etapie zostanie utworzony zbiór identyfikatorów jednostek, spełniający zadane kryterium albo określony przez użytkownika w inny sposób. Zbiór ten będzie następnie sterować czynnościami drugiego etapu — odczytywaniem danych jednostkowych i ich agregacją.

Rozwiązanie takie jest optymalne w warunkach korzystania z SZBD RODAN, który dysponuje buforem o pojemności jednej strony dla każdego obszaru. W tych bowiem warunkach liczba zmian zawartości bufora (i w związanych z tym długotrwałych przesłań pomiędzy PAO i urządzeniami pamięci zewnętrznej) zależy jedynie od prawdopodobieństwa znalezienia potrzebnej danej w buforze.

Wprowadzenie dwufazowego wyszukiwania danych pozwala skuteczniej niż przy wyszukiwaniu jednofazowym optymalizować proces przetwarzania.

System przetwarzania zewnętrznego

jest z założenia mniej spójnym fragmentem oprogramowania banku danych BIS. Obejmuje on mianowicie oprogramowanie ściśle użytkowe, a zatem mocno związane z dziedziną, w której zostanie wykorzystany bank danych. Opierając się na strukturze systemu przetwarzania wewnętrznego, można jednak określić tu również dwa podstawowe moduły: **moduł zasilania** i **moduł wyjścia**.

Zadaniem modułu zasilania jest przygotowanie danych przeznaczonych do wprowadzenia do bazy danych. W zakres tej czynności wchodzi m.in. następujące funkcje:

- kontrola poprawności danych (formalna i rachunkowa)
- kontrola kompletności danych, gdy kryterium kompletności nie może być wyprowadzone z zasobów banku danych
- konwersja danych według wymagań modułu aktualizacji systemu przetwarzania wewnętrznego.

Zadaniem modułu wyjścia jest z kolei udostępnienie danych w założonej formie. Można tu wyróżnić:

- emisje wydawnictw o charakterze tabelarycznym
- udostępnienie zestawień sformułowanych i przetworzonych według różnych kryteriów
- tworzenie maszynowo zorientowanych zbiorów danych dla zasilania innych systemów informatycznych, statystycznych i resortowych.

W celu realizacji tych modułów szeroko wykorzystane będzie istniejące już oprogramowanie użytkowe i systemowe. Szczególne miejsce zajmują tu generator tablic TPL,

standardowe programy sortowania i podprogramy statystyczno-matematyczne. Ponadto zostaną użyte elementy oprogramowania różnych, eksploatowanych w GUS systemów przetwarzania, oczywiście po przeprowadzeniu odpowiednich prac adaptacyjnych. Własne programy, tworzone w razie niezbędnej konieczności, cechować się będą dużą uniwersalnością. Osiągnięte to zostanie przez ich parametryzację, kodowanie w języku wysokiego rzędu, wykorzystanie możliwości generacyjnych preprocesora PL1 itp.

Bez ryzyka można stwierdzić, że nie ma innej możliwości dla rozwoju systemów przetwarzania danych niż coraz szersze wykorzystywanie uniwersalnych pakietów programowych typu SZBD. Przesądzą o tym przede wszystkim koszty realizacji i utrzymania indywidualnego oprogramowania, które zapewniłoby systemowi przetwarzania własności równorzędne z osiąganymi przy zastosowaniu pakietów standardowych. Warunkiem uwieńczonego powodzeniem wdrożenia SZBD jest jednak właściwe wykorzystanie jego własności na potrzeby konkretnego zastosowania. W przypadku sieciowego SZBD kluczowym problemem jest określenie struktury bazy danych, zapewniającej systemowi przetwarzania odpowiednie walory użytkowe.

Eksperyment z bazą danych BASTA wykazał ograniczoną efektywność intuicyjnego podejścia do tego problemu, polegającego na odwzorowaniu w schemacie bazy danych obiektów i relacji opisywanej rzeczywistości. Zakres stosowania tej metody jest ograniczony do małych i nieskomplikowanych baz danych. Baza danych przedstawionego tu banku BIS powstała w wyniku podjęcia decyzji projektowych, których celem było skrócenie ścieżek selekcji danych i zmniejszenie utrzymywanego przez SZBD stopnia ich uporządkowania. Do najważniejszych z nich należą:

- 1) odejście od skomplikowanej struktury głównego zbioru danych
- 2) podział głównego zbioru danych na jednostki tematyczno-czasowe
- 3) wydzielenie wspólnego dla całej bazy danych masywu katalogowego
- 4) maksymalne rozszerzenie możliwości wewnętrznego systemu przetwarzania.

Dzięki temu osiągnięto także znaczne korzyści w produkcji oprogramowania użytkowego. Korzyści te uzyskano również dzięki temu, że SZBD przejmuje na siebie wszelkie czynności związane z operowaniem na fizycznej strukturze danych, co umożliwia zawężenie tematyki oprogramowania wykonywanego samodzielnie do specyficznych problemów danego zastosowania. Udogodnienia techniczne zawarte w SZBD, polegające np. na automatycznej kontroli i konwersji danych lub automatycznej aktualizacji wyspecyfikowanych danych, mają również wpływ na korzystną ocenę warunków realizacji oprogramowania użytkowego.

LITERATURA

- [1] Raport Grupy Roboczej ds. Baz Danych CODASYL. OBRI Warszawa 1977
- [2] CODASYL. Język Opisu Danych. OBRI Warszawa 1977
- [3] Bank Danych Statystycznych w technologii DBTG CODASYL. „Systemy Informatyczne” nr 2/78
- [4] Lasota M.: Metoda graficznego opisu bazy danych. „Wiadomości Statystyczne” nr 9/78
- [5] Lasota M., Lipkowska E.: Statistical Data Bank in DBTG CODASYL Technology. Proceedings of the ISIS 78 SEMINAR (w przygotowaniu)

Patentowanie oprogramowania komputerów

Czy projektanci nowych oryginalnych rozwiązań mają prawo do ochrony patentowej swych programów? Pytanie to nie jest nowe. Dziś po latach wielu dyskusji prawnych i technicznych w USA, trzeba powiedzieć, że: tak, ale gdy zachodzą określone okoliczności. Decydujący jest tu warunek, aby wynalazek był zakwalifikowany do kategorii „sztuka użytkowa”.

Urząd ochrony praw autorskich (Copyright Office) stoi dotąd na stanowisku, że nie warto rejestrować programów komputerowych. Prawdopodobnie dlatego, że liczba rozstrzygnięć sądowych jest bardzo mała. Mimo to w urzędzie zarejestrowano już około 1300 programów komputerowych.

Prawo autorskie nie nadąża za dzisiejszym rozwojem informatyki. Zabrania ono bezprawnego przedruku niezależnie od treści dzieła, a jedynym kryterium jest jego oryginalność. Plagiator czy nieuczciwy konkurent może tak długo korzystać z cudzej idei, dopóki nie użyje specyficznych wyrażań, a więc zastrzeżonej formy.

Prawo patentowe natomiast nastawione jest na ochronę urządzeń i metod będących urzeczywistnieniem idei. Jeżeli wynalazek dotyczy wyłącznie obszaru „czystej nauki”, a nie „sztuki użytkowej”, wówczas zostanie uznany za „niestatutowy”, a więc nie nadający się do opatentowania.

Obydwa te prawa dość dobrze służą swym celom w obszarze tradycyjnych zastosowań, zawodzą zaś w przypadku systemów informatycznych. Komputer jest bowiem organiczną jednością programu i sprzętu realizującego ten program. Istotą programu są „abstrakcyjne intelektualne koncepty”, a więc konstrukcje myślowe. Jest to klasyczny przykład osiągnięć na gruncie teorii, a więc osiągnięć nie mających, jak dotąd, zdolności patentowej.

Oto kilka przykładów. W każdym z przytoczonych niżej przypadków amerykański Patent and Trademark Office (Urząd ds. patentów i znaków fabrycznych — PTO) wymagał, aby wniosek o udzielenie patentu nie pozostawiał wątpliwości co do tego, że przedmiot wynalazku ma wartość użytkową, jest rozwiązaniem nowym i „nie oczywistym”. Odrzucano podania, w których przedmiot wynalazku nie spełniał tych wymogów lub nie odpowiadał przyjętej klasyfikacji. Każdy z podanych przypadków był przyczyną postępowania odwoławczego, jednak ze zmiennym szczęściem.

Gary Benson i Arthur Tabbot wynaleźli metodę przetwarzania sygnałów dziesiętnych kodowanych dwójkowo na sygnały kodowane w innych układach. Wniosek o udzielenie patentu został odrzucony ponieważ „metoda przetwarzania sygnałów” nie figurowała w klasyfikacji patentowej. Wynalazcy odwoływali się do coraz wyższych instancji aż do Sądu Najwyższego, który niestety podtrzymał stanowisko urzędu patentowego (PTO). Podobnych spraw było bardzo dużo. Jeżeli tylko w zgłoszeniu patentowym występowały wzory matematyczne, równania czy procesy algorytmiczne, patentu z reguły nie udzielano.

Glenn Chatfield rozwiązał zagadnienie uniknięcia kolizji dwu lub więcej programów w systemie wieloprogramowym, gdy żądają one w tym samym czasie tych samych urządzeń systemu komputerowego. Po odrzuceniu wniosku przez

urząd patentowy wynalazca odwołał się do sądu i proces wygrał. Sąd apelacyjny nie zgodził się ze stanowiskiem urzędu patentowego, że wynalazek nie nadaje się do opatentowania jeżeli nowość zawarta jest w algorytmie programu przeznaczonego dla maszyny cyfrowej.

W podobny sposób, na drodze sądowej, otrzymał patent Murray Deutsch. Wynalazca systemu sterowania i optymalizacji pracy zakładów wielooddziałowych i terenowo rozproszonych, jak na przykład rafinerie ropy naftowej, wystąpił także na drogę sądową.

Dean Christensen wynalazł sposób określania porowatości warstw podpowierzchniowych (*porosity of subsurface formations*). Urząd patentowy zignorował wynalazek Christensena (jakoby był bez wartości praktycznej), a wszelkie apelacje i w tym przypadku nie pomogły.

Gerald Waldbaum opracował sposób naliczania czasu zajętości i niezajętości „numerów” linii abonenckich w centralach telefonicznych poprzez pewną nową organizację bitów w poszczególnych wyrazach przesyłanych informacji. Niestety sąd uznał ten przypadek za analogiczny do przypadku pana Christensena i przyznał rację urzędowi patentowemu.

Jednakże dwie dalsze sprawy przyniosły postęp w rozumowaniu sędziów.

Dale Flook wynalazł sposób kontrolowania jednego bądź kilku parametrów w procesie katalizacyjnej przemiany węglowodorów i jednocześnie sygnalizowania stanów zagrożenia, gdyby proces odbiegł od normy. Rozpoznanie stanu zagrożenia następowało w wyniku okresowo dokonywanego rachunku w oparciu o pewne równanie matematyczne, wyprowadzone przez Flooka, i wartości pomiaru parametrów procesu aktualnych i uprzednio dokonanych, przy czym cały rachunek wykonywał komputer.

Urząd patentowy, dopatrzwszy się przedmiotu wynalazku w równaniu matematycznym, oddalił wniosek o udzielenie patentu. Natomiast sąd stanął na stanowisku, że Flook chciał opatentować nie sam algorytm, ale szczególny sposób praktycznego użytkowania tego algorytmu, sposób powodujący istotną modyfikację znanego wcześniej procesu technologicznego i nakazał wydanie patentu.

Drugi przykład dotyczy wynalazku Donalda Richmana. Wynalazł on metodę obliczania korekcji katowej dla lotniczego pokładowego radaru wykorzystującego efekt Dopplera, umożliwiającą wyznaczenie średniej szybkości wznośzenia się lub zniżania się samolotu. Ponieważ metoda ta w istocie swej sprowadzała się do działań matematycznych, sąd orzekł, że nie ma ona zdolności patentowej.

Tak więc metoda Flooka nadawała się do opatentowania ponieważ mogła być zakwalifikowana jako „sztuka użytkowa”, a metoda Richmana — nie, ponieważ uznano, że należy do „czystej nauki”. I bądź tu mądry.

W nowym amerykańskim prawie autorskim, obowiązującym od 1 stycznia 1978 r., znalazła się jednak wzmianka, która mogłaby wskazywać na tendencję zmierzającą do objęcia programów komputerowych ochroną prawną.

Oprac. według „Informacji Adresowanej”, OBRI 1978

Bardzo przepraszamy za błąd w tytule artykułu Z. Czyrka i K. Gerwina, opublikowanego w numerze 1/79 **INFORMATYKI**. Powinno być: „System obsługi bazy kontenerowej (nie komputerowej) z ruchomymi terminalami i radiową transmisją danych”.

