

biuletyn informacyjny

2
'81

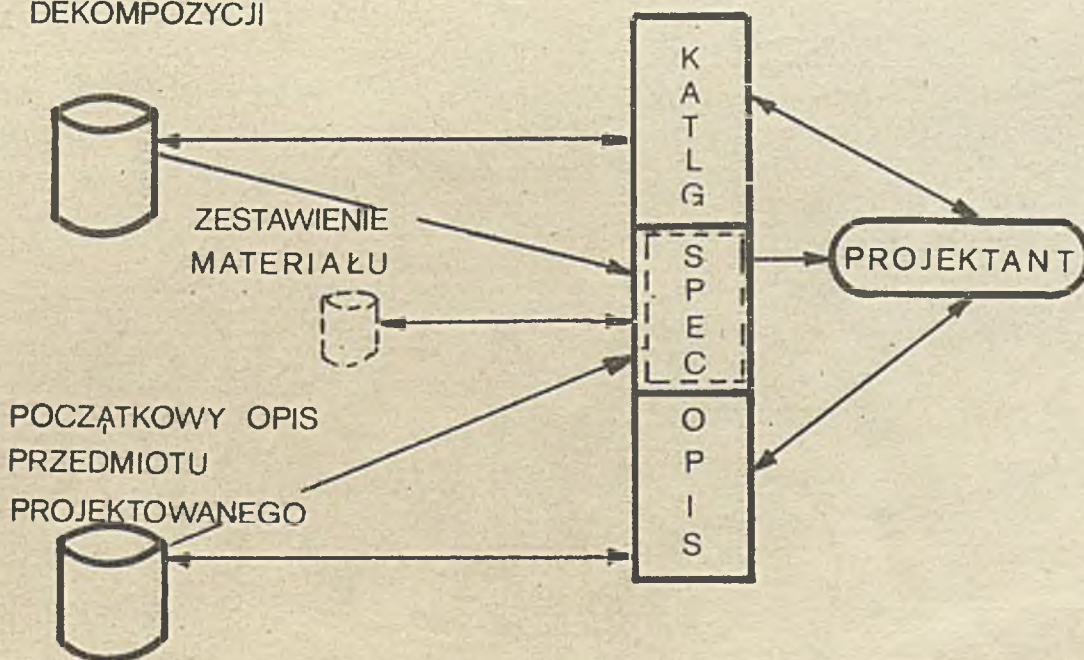


OBIKTOWE
SYSTEMY
KOMPUTEROWE



9.3057/81

WZORY
DEKOMPOZYCJI



Rysunek na okładce: System wspomagający projektowanie / J Mocała /

Druk okładki IMM zam. 85/81 n. 660.

mgr inż. Grzegorz ZIEMBICKI

Biuro Generalnych Dostaw

Pracownia Projektowa



Minikomputerowy system alarmowy

● Możliwości realizacji

Gwałtowny, ogólnosiwiatowy rozwój technologii ostatnich lat w dziedzinie mikroelektroniki - a zwłaszcza opanowanie na skalę przemysłową produkcji seryjnej układów scalonych średniej (MSI) i dużej (LSI) skali integracji, umożliwiające rozwój techniki w zakresie budowy mini- i mikroprocesorów, pozwolił na - oparte na przesłankach ekonomicznych - rozszerzenie obszaru zastosowań systemów komputerowych na sferę pomiarów, automatyki i kontroli.

Jedną z dziedzin wchodzących w skład ww. obszarów aplikacji stanowią także mini/mikrokomputerowe systemy alarmowe o różnym przeznaczeniu, jak np. opisany dalej system alarmów przeciwpożarowych.

Nowoczesne mini/mikrokomputerowe systemy zabezpieczenia przeciwpożarowego wyposażone w wyspecjalizowane urządzenia zrealizowane na układach MSI i LSI, jak np. dalej opisany koncentrator sygnałów alarmowych umożliwiają w szerszym niż dotychczas zakresie rozwiązanie problemów automatyzacji całego kompleksu zagadnień z zakresu dozoru obiektów, wczesnego wykrywania i lokalizacji źródeł alarmów, powiadamiania o nich ośrodków dyspozycyjnych.

Tematem niniejszego opracowania będzie propozycja realizacji minikomputerowego systemu alarmowego sygnalizacji pożarów, opracowywanego w Pracowni Projektowej Systemów Minikomputerowych BGD Centrum-MERA w Warszawie, a opartego na:

- wykorzystaniu istniejącego krajowego sprzętu z zakresu dozoru i sygnalizacji pożarów, sprzężonego z minikomputerowym systemem MERA-400 w wersji pozastandardowej, rozszerzonej o wyspecjalizowane urządzenia pozwalające na zastosowanie go w wymienionym zakresie;
- oryginalnych opracowaniach w zakresie
 - systemowych urządzeń wyspecjalizowanych typu układu automatycznego załączania/wyłączania systemu minikomputerowego (AZ/WSM)[1], koncentratora sygnałów alarmowych (KSA)[2];
 - oprogramowania systemowego "system zabezpieczenia przeciwpożarowego", opracowanego na podstawie "uniwersalnego pakietu prowadzenia kartotek dyskowych - ADA2"[3].

● Organizacja minikomputerowych systemów alarmowych

Mając na uwadze przesłanki ekonomiczne związane z organizacją techniczną, eksploatacją i obsługą serwisową, należy przyjąć, że struktura organizacyjna minikomputerowego systemu sygnalizacji pożarów będzie uzależniona głównie od wielkości dozoruwanej przez niego jednostki w rodzaju:

- pojedynczego obiektu typu biurowca, hotelu, muzeum itp.,
- zespołu obiektów typu zakładu produkcyjnego, kopalni, stoczni, kombinatu przemysłowego itp.,
- obszaru administracyjnego, jak np. miasto, powiat, województwo.

Powyższa zależność będzie decydująca przy woborze jednego z wariantów rozwiązania organizacyjnego systemu, które z kolei można podzielić zasadniczo na dwa typy:

- systemy autonomiczne - przeznaczone do dozoru pojedynczych obiektów lub ich zespoły, z własną służbą straży pożarnej i gdzie ze względu na powyższe i na specyfikę lokalną wystarczającym zabezpieczeniem będzie autonomiczny system alarmowy; wspomniane służby są odpowiednio wyspecjalizowane w przeciwdziałaniu różnym stanom zagrożenia, zależnym od warunków lokalnych występujących np. w górnictwie, technice jądrowej, petrochemii, chemii, na jednostkach pływających itp.;
- systemy satelitarne - przeznaczone do dozoru tych obiektów lub ich zespołów, nie mających własnych służb pożarowych a leżących w obrębie obszaru administracyjnego znajdującego się pod kontrolą centralnego systemu alarmowego. Centralny ośrodek dyspozycyjny zawierający między innymi własny bank danych o poszczególnych dozorowanych przez system obiektach, znajduje się w rejonowej (miejskiej, wojewódzkiej) komendzie straży pożarnej.

Tematem dalszego ciągu opracowania będą właśnie systemy satelitarne. Obejmują one liczną klasę minikomputerowych systemów alarmowych, w których zawsze występuje konieczność przesyłania meldunków alarmowych liniami telekomunikacyjnymi, przy czym może to być realizowane bądź automatycznie, bądź za pośrednictwem personelu obsługi.

● Proponowane rozwiązanie techniczne systemu satelitarnego

Przedstawiony niżej minikomputerowy satelitarny system alarmowy, przeznaczony do automatycznej sygnalizacji pożarów, opracowywany w Pracowni Projektowej Systemów Minikomputerowych BGD Centrum MERA na zlecenie Wojewódzkiej Komendy Straży Pożarnej w Elblągu, ma objąć swoim zasięgiem obszar całego województwa elbląskiego. Proponowane rozwiązanie opiera się całkowicie na istniejącym sprzęcie krajowym. Rozwiązanie to, za pomocą poszczególnych urządzeń systemowych gwarantuje realizację następujących założeń na wstępie funkcji:

- dozowanie,
- sygnalizacja pożaru,
- powiadomienie - z możliwością podania adresu alarmowego w sposób automatyczny, zdalnie, z lokalnego układu KSA, poprzez linię modemową do jednostki centralnej (JC) systemu,
- otrzymanie z banku danych przez ośrodek dyspozycyjny kierujący akcją przeciwpożarową informacji dotyczących
 - charakterystyki zagrożonego obiektu,
 - planu akcji przeciwdziałania,
 - optymalnych dróg dojazdowych
 - oraz wykazu sił i środków potrzebnych do przeciwdziałania pożarowi

Przedstawiony na rys.1 minikomputerowy system alarmowy o strukturze satelitarnej przeznaczony

D W U M I E S I Ę C Z N I K

Wydaje:

CENTRUM NAUKOWO-PRODUKCYJNE TECHNIK KOMPUTEROWYCH I POMIARÓW
I N S T Y T U T M A S Z Y N M A T E M A T Y C Z N Y C H
Branżowy Ośrodek Informacji Naukowej Technicznej i Ekonomicznej

KOMITET REDAKCYJNY

dr inż. Stanisława BONKOWICZ-SITTAUER, doc.mgr Jan BOROWIEC,
mgr Cezary DZIADOSZ /sekretarz redakcji/
doc.dr inż. Jan ŁYSKANOWSKI, doc.dr hab.inż. Stanisław MAJERSKI,
doc.dr inż. Henryk ORŁOWSKI /redaktor naczelny/
dr inż. Piotr PERKOWSKI

Opracowanie redakcyjne: mgr Hanna DROZDOWSKA

Opracowanie graficzne: Barbara KOSTRZEWSKA

Adres redakcji: ul. Krzywickiego 34, 02-078 Warszawa
tel. 21-84-41 w.244 lub 28-37-29

Biuletyn Informacyjny **OBIEKTOWE SYSTEMY KOMPUTEROWE**

Rok XIX

Nr 2

1981

Spis treści

Содержание

Contents

ZIEMBIŃSKI G.: Minikomputerowy system alarmowys. 3

z cyklu: Komputeryzacja projektowania

BONKOWICZ-SITTAUER S.: Problemy oprogramowania projektującegos. 9

MOCALA J.: Wprowadzenie do systemu MUMPSs.23

MOCALA J.: Automatyzacja procesu projektowania. Program SPEC dla EMC MERA 400.....s.41

SPRAWOZDANIA z konferencjis.51

Założenia i warunki świadczenia usług w zakresie generacji testów dla zakładów Zjednoczenia MERAs.57

ЗЕМБИЦКИ Г.: Миникомпьютерная противопожарная сигнальная системас. 3

из цикла: Компьютеризация проектирования

БОНКОВИЧ-СИТТАУЭР С.: Проблемы проектирующего программного обеспеченияс. 9

МОЦАЛА Й.: Введение в систему MUMPSс.23

МОЦАЛА Й.: Автоматизация процесса проектирования. Программа SPEC для ЭВМ MERA 400 ...с.41

ОТЧЕТЫ с конференциис.51

ZIEMBIŃSKI G.: Minicomputer alarm systemp. 3

from the series: Computerisation of programming

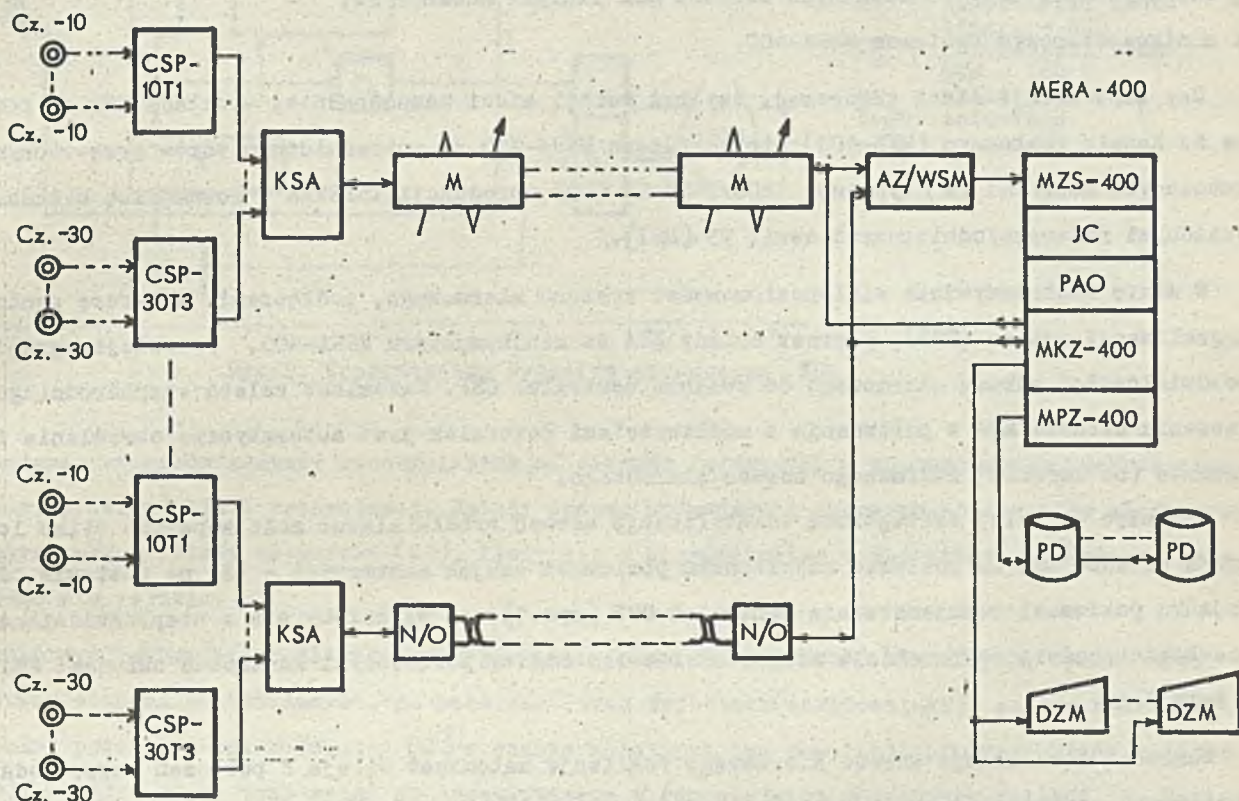
BONKOWICZ-SITTAUER S.: Problems of designing softwarep. 9

MOCALA J.: Introduction to the MUMPS systemp.23

MOCALA J.: Automatization of the design process. SPEC program for MERA 400 computerp.41

The conference REPORTS ...p.51

jest do dozorowania obszaru administracyjnego, w obrębie którego poszczególne kontrolowane obiekty są (lub zostaną) wyposażone w lokalne obiektowe sieci alarmowe.



Cz - czujki alarmowe
 CSP - centralka sygnalizacji pożaru
 KSA - koncentrator sygnałów alarmowych
 M - modem
 DZM - drukarka mozaikowo-znakowa z klawiaturą

N/O - układ nadawczo/odbiorczy linii
 AZ/WSM - układ automatycznego załączania/wyłączenia systemu minikomputerowego
 PD - pamięć dyskowa

Rys. 1. Minikomputerowy system alarmowy w wersji satelitarnej

Sieci lokalne będą zorganizowane w dwu wersjach, odpowiednio na bazie urządzeń:

- w wersji A - czujek alarmowych³⁶, reagujących na przekroczenie dopuszczalnego poziomu dymu lub temperatury, reagujących na ogień oraz na pochodne tych zjawisk; czujniki te należą do typoszeregu 10 i 30, produkowane są przez Zakłady Urządzeń Dozymetrycznych POLON w Bydgoszczy; czujniki te będą podłączone do
 - centralek sygnalizacji pożarów TELSAP typu CSP-10T1, CSP-30T3 produkcji Zakładów Telekomunikacyjnych TELKOM-TELF A w Bydgoszczy;
- w wersji B - czujek alarmowych - jw., podłączonych do
 - koncentratorów sygnałów alarmowych (KSA).

³⁶ Termin używany przez producenta

Urządzenia te połączone są dalej w strukturach systemu:

- zbiorczo (rys.1), za pośrednictwem lokalnych sieci przewodowych poprzez nadrzędne układy KSA i dalej liniami modemowymi,
 - bezpośrednio, z wyjść lokalnych układów KSA liniami modemowymi,
- z minikomputerowym systemem MERA-400.

Czy to w wersji sieci zbiorczej, czy też wersji sieci bezpośredniej - układy KSA są podłączone do kanału znakowego (MKZ-400) minikomputera MERA-400 za pośrednictwem torów przewodowych zakończonych modemami (M) typu np. 1200/2400 EC 8013, produkcji TELETRA - Poznań lub układami scalonymi nadawczo/odbiorczymi serii 75 (MSI).

W miarę rozbudowywania wielopoziomowości systemu alarmowego, podłączenie zbiorcze centralek sygnalizacji pożaru (CSP), poprzez układy KSA do minikomputera MERA-400, spowoduje ograniczenie rozdzielczości adresu alarmowego do poziomu centralek CSP. Natomiast zaletą bezpośredniego podłączenia układów KSA w porównaniu z możliwościami centralek jest automatyczne określanie i przekazanie (na żądanie) dokładnego adresu alarmowego.

Tak więc bardziej szczegółową identyfikację adresu źródła alarmu może zapewnić tylko lokalizacja układów KSA na poziomie najbliższym poziomowi czujek alarmowych - tj. na poziomie odpowiadającym poziomowi rozmieszczenia centralek CSP (rys.1), co wiązałoby się z nieprzewidzianą obecnie koniecznością retransmisji adresu alarmowego między połączonymi kaskadowo układami KSA w wersji A.

Dokładną lokalizację adresu alarmowego realizuje natomiast wersja B połączeń bezpośrednich na poziomach: czujki alarmowe - koncentrator (KSA) i koncentrator - minikomputer MERA-400. Wadą jej (w większym stopniu niż w wypadku połączeń wersji A) jest ograniczona liczba miejsc adresowych kanału znakowego (MKZ-400) minikomputera (maksimum 3 kanały, 8 miejsc adresowych każdy), ograniczająca możliwości rozbudowy systemu alarmowego.

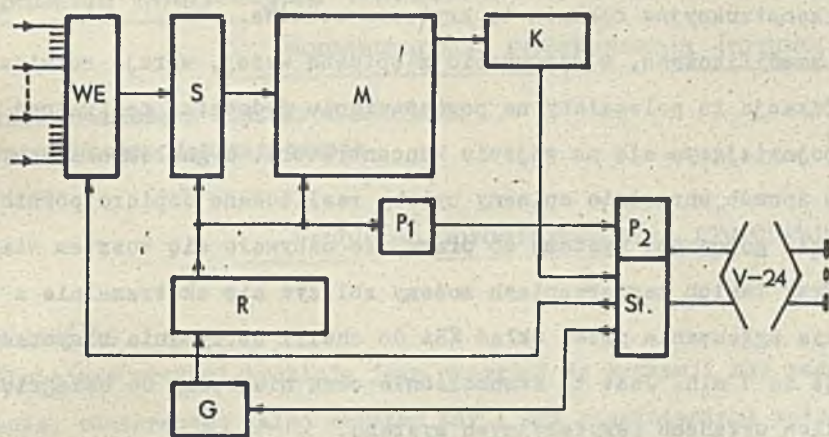
Lepsze perspektywy może stworzyć w przyszłości w tym względzie kanał multipleksorowy MERA-400, dysponujący osiemdziesięcioma wejściami modemowymi.

● Koncentrator sygnałów alarmowych KSA i układ automatycznego załączania/wyłączania systemu minikomputerowego AZ/WSM

Elementem wiążącym każdy lokalny system alarmowy na poziomie pojedynczego obiektu, zespołu obiektów, czy też pewnego rejonu dozorowanego obszaru będzie zawsze dla obu przedstawionych wersji połączeń A i B układ koncentratora sygnałów alarmowych (KSA), którego schemat blokowy przedstawia rys.2.

W przyjętym rozwiązaniu układ KSA ma maksymalnie 256 wejść (WE) adresowalnych, umożliwiających przyjmowanie sygnałów alarmowych bezpośrednio z czujek alarmowych lub pośrednio z czujek poprzez centralki CSP. Każdemu potencjalnemu źródłu alarmu przypisane jest odpowiednie wejście (WE) o określonym numerze, któremu w zbiorze informacji banku danych systemu odpowiada adres rzeczywisty obiektu w terenie.

Podobnie, jak w centralkach CSP, tak i w koncentratorach KSA układy wejściowe umożliwiają



- WE - układy wejściowe
- S - selektory
- R - rejestry
- G - generator
- M - pamięć typu RAM
- K - komparator
- P₁ - przetwornik kodu:
binary - BCD
- P₂ - przetwornik kodu:
BCD - ISO 7
- St - układ sterujący
- V-24- interfejs

Rys.2. Koncentrator sygnałów alarmowych KSA

rozróżnianie sygnałów alarmów rzeczywistych od alarmów fałszywych - spowodowanych uszkodzeniem (zwarcie, przerwa) linii przewodowej. Dalszy proces sprawdzania poprawności, sygnałów alarmowych przyjętych przez układy wejściowe (WE), łączy się z zapamiętaniem w rejestrach (R) adresu zgłaszającego się wejścia.

W celu wyeliminowania możliwości przyjmowania fałszywych zgłoszeń alarmowych, wymuszonych chwilowymi stanami niestabilnymi, na poszczególnych wejściach (zakłóceniami), są one kilkakrotnie próbkowane przez generator lokalny (G) w czasie kolejnych ich komutacji poprzez układy selektorów (S). Kolejne stany tych wejść są zapamiętywane w odpowiednich rejestrach pamięci M. Pamięć (M) typu RAM została podzielona organizacyjnie na szereg rejestrów przyporządkowanych poszczególnym wejściom (WE) koncentratora KSA.

Każde z wejść (WE) jest próbkowane w czasie komutacji przez czas potrzebny do dokonania odczytu wartości odpowiadającego mu rejestru pamięci. Wartość końcowa tego rejestru badana jest w układzie komparatora (K) pod kątem liczby zapamiętanych potórzeń stanów zgłoszeń alarmowych. Jeżeli układ komparacji (K) wykryje w badanym rejestrze pamięci stan odpowiadający kodem stanowi alarmu, to inicjuje on generację sygnału alarmu na linię modemową. Sygnał ten poprzez układ sterujący (St) koncentratora KSA uruchamia zdalnie układ automatycznego załączania/wyłączania systemu minikomputerowego (AZ/WSM). Z kolei układ AZ/WSM stawia w stan gotowości do pracy system MERA-400 oraz inicjuje dwustronną transmisję informacji.

Na pytanie jednostki centralnej (JC) systemu o adres źródła alarmu koncentrator, do którego skierowana jest transmisja, przesyła informację zawierającą żądany adres zgłaszającego się wejścia (WE). Informacja ta jest uprzednio przetworzona w układach przetworników (P₁ i P₂) kodów, tj. z kodu binarnego - na kod BCD oraz z BCD na kod ISO-7.

Przyjmując czas potrzebny na wypracowanie sprawdzonego sygnału alarmowego, zgłoszenie alarmu, dwustronną transmisję informacji i podanie pełnej charakterystyki zagrożonego obiektu na $\leq 0,5$ min. oraz czas osiągnięcia gotowości systemu MERA-400 do pracy = 1 min. należy przyjąć, że w czasie $\cong 1,5$ min. Straż pożarna powinna dysponować kompletem informacji potrzebnych do rozpoczęcia akcji przeciwpożarowej.

Opisane uprzednio rozwiązanie konstrukcyjne spełnia te kryteria czasowe.

Można natomiast przyjąć nieco zmodyfikowaną, w porównaniu z opisaną wyżej, wersję rozwiązania organizacyjnego układu KSA. Modyfikacja ta polegałaby na powiadamianiu jednostki centralnej JC systemu MERA-400, już pierwszym pojawiającym się na wejściu koncentratora, sygnałem alarmowym. Badanie prawdziwości zgłoszenia w sposób uprzednio opisany byłoby realizowane dopiero później, w takim jednak czasie, aby osiągnięcie gotowości systemu do pracy nie odbywało się kosztem wiarygodności zgłoszenia alarmowego. Przy takich zamierzeniach możemy zbliżyć się ekstremalnie z czasem upływającym od chwili przyjęcia zgłoszenia przez układ KSA do chwili otrzymania z systemu minikomputerowego pełnej informacji do 1 min. Jest to równocześnie czas niezbędny do osiągnięcia pełnej gotowości do pracy wszystkich urządzeń peryferyjnych systemu.

Zostałoby to osiągnięte kosztem ewentualnej rejestracji również fałszywych sygnałów alarmowych, które za pośrednictwem układu AZ/WSM włączałyby i stawiały w stan gotowości system MERA-400. Stan gotowości systemu do pracy byłby utrzymywany do czasu otrzymania przez JC systemu, informacji z koncentratora (KSA) o fałszywości alarmu, a ściślej o braku potwierdzenia tego alarmu adresem zgłaszającego się wejścia (WE).

Zdalne, automatyczne załączanie/wyłączanie minikomputerowego systemu alarmowego realizowane jest za pośrednictwem skorelowanych czasowo sekwencji sygnałów. Sygnały te podawane są z układu AZ/WSM do modułu zasilacza sekwencyjnego (MZS-400) MERA-400 oraz do poszczególnych urządzeń peryferyjnych; tj. pamięci dyskowych (PD) typu MERA-9425, drukarek mozaikowo-znakowych z klawiaturą (DZM) typu DZM180 + KSR i pozwalają na uniknięcie zbędnej eksploatacji zespołów mechanicznych powyższych urządzeń, występującej przy pracy non-stop w czasie dozoru, uruchamiając je dopiero z chwilą nadejścia sygnału alarmowego.

Uwagi

Przedstawiona wyżej propozycja rozwiązania technicznego minikomputerowego systemu alarmowego do osłów sygnalizacji zagrożenia pożarowego stanowi, jak już było wspomniane, opracowanie samodzielne.

W dostępnej literaturze nie udało się trafić na materiały dotyczące powyższego zagadnienia, mimo informacji o wytwarzaniu systemów do celów zabezpieczenia przeciwpożarowego w najbardziej rozwiniętych przemysłowo krajach.

Literatura wykorzystana przy pracy nad projektem

- [1] Automatyczne załączanie/wyłączanie systemu minikomputerowego. Pracownia Projektowa BGD Centrum-MERA, 1980
- [2] Koncentrator sygnałów alarmowych. Pracownia Projektowa BGD Centrum-MERA, 1981
- [3] Uniwersalny pakiet prowadzenia kartotek dyskowych - ADA2. Pracownia Projektowa BGD Centrum-MERA, 1980
- [4] Urządzenia sygnalizacji pożaru. ZUD POLON Bydgoszcz, 1979
- [5] Centralka sygnalizacji pożaru. TELKOM-TELFA Bydgoszcz, 1977
- [6] Iskrobezpieczny elektryczny system alarmowo-pożarowy TELSAP-1. ZUD POLON, 1979
- [7] Minikomputerowy system MERA-400. CNPTK i P MERA, 1980

Komputeryzacja projektowania inżynierskiego

dr inż. Stanisława BONKOWICZ-SITTAUER
Instytut Maszyn Matematycznych

Problemy oprogramowania projektującegoUwagi ogólne

Niniejszy artykuł traktuję jako materiał do dyskusji nad zagadnieniami wspomaganego projektowania, oznaczanego dalej skrótem KWP i nad zagadnieniami automatyzacji poszczególnych prac inżynierskich oznaczanych jako API.

Na wstępie należy zaznaczyć, że w zakresie projektowania inżynierskiego można automatyzować, czy też komputeryzować jedynie poszczególne czynności. Natomiast cały proces projektowania konkretnych obiektów inżynierskich, zawiera liczne elementy oraz fragmenty n i e k o m p u t e r y z o w a l n e .

Dlatego mówi się o automatyzacji prac inżynierskich (API) oraz o komputerowo wspomaganym projektowaniu (KWP).

Stopień komputeryzowalności poszczególnych procesów projektowych zależy od merytorycznej treści tych procesów i ich formy. Oznacza to, że dobór konkretnych zestawów sprzętu i oprogramowania dla konkretnych procesów projektowych może być dokonany dopiero w wyniku analizy danych procesów. Równocześnie jednak nowe możliwości sprzętowo-programowe oddziałują nieraz bardzo silnie i głęboko na charakter poszczególnych procesów projektowych. Dlatego ważne jest, aby planując zaspokojenie krajowych potrzeb w zakresie systemów KWP nie projektować sztywnych konfiguracji sprzętu i oprogramowania, aby pewien repertuar możliwości, z których byłaby kompletowana konkretna konfiguracja uwzględniająca specyfikę potrzeb danej organizacji projektowej.

Przystępując do planowania przedsięwzięć mających na celu uporządkowanie prac nad oprogramowaniem dla KWP w ogólności, a szczególnie dla automatyzacji poszczególnych prac inżynierskich, należy przede wszystkim dokonać identyfikacji rodzajów klas samego oprogramowania bezpośrednio wykorzystywanego w procesie projektowym, a zwanego dalej oprogramowaniem projektującym oraz identyfikacji wszelkich narzędzi programistycznych, które są potrzebne do powstawania oprogramowania projektującego.

Taka identyfikacja poszczególnych typów, klas i rodzajów oprogramowania dla KWP pozwoli na ocenę, kto, w jakim trybie i w jakim zakresie powinien prowadzić prace zarówno nad dokładniejszą dalszą identyfikacją tego oprogramowania, jak i nad tworzeniem poszczególnych składników takiego oprogramowania.

Należy też zwrócić uwagę, iż poza badaniami nad zasadniczą strukturą i przebiegiem procesu tworzenia oprogramowania projektującego (proces TOP), należy studiować różnorodne problemy towarzyszące temu procesowi. Problemy te nieraz wymagają jeszcze podstawowych badań metodycznych. I tak należy m.in.:

• doskonalić formy dokumentacji opracowania dostosowując je do potrzeb użytkowników-projektan-

- tów lub twórców kolejnych warstw programowych,
- poszukiwać sposobów i uściślać kryteria oceny jakości oprogramowania projektującego
 - wypracowywać właściwe rozwiązanie licznych prawnych aspektów związanych z wprowadzeniem KWP, a mianowicie:
 - ponoszenie odpowiedzialności za jakość lub tylko poprawność wykonywanych danym programem projektów
 - sposoby atestowania programów,
 - autorstwo projektu a autorstwo oprogramowania
 - badać i wypracowywać sposoby rozpowszechniania informacji o istniejącym i powstającym oprogramowaniu projektującym oraz stwarzać platformy wymiany poglądów i ocen na temat oprogramowania projektującego.

Tej klasy problemy, aczkolwiek mogą się wydawać drugorzędne, nie powinny jednak umykać z pola widzenia środowiska zainteresowanego zagadnieniami KWP, gdyż na czas nie podjęte, mogą się stać źródłem poważnych zahamowań rozpowszechniania metod komputerowych w projektowaniu, a to może z kolei odbić się na opóźnieniach rozwoju całej gospodarki.

W chwili obecnej fakt możliwości i celowości stosowania KWP w ogóle oraz API w odniesieniu do poszczególnych fragmentów procesów projektowych jest w zasadzie uznawany powszechnie przez projektantów. I jeżeli w kraju powszechność KWP nie jest jeszcze zadowalająca, to głównie z powodu braków sprzętu i oprogramowania.

Brakuje bowiem nawet tak podstawowego obecnie dla API sprzętu, jak kasety dyskowe, czy plottery. W konsekwencji brakuje również podstawowego oprogramowania dla potrzeb KWP typu pomocnicze biblioteki matematyczno-techniczne, programowe mechanizmy gospodarki zbiorami, pakiety procedur dla generowania dokumentacji projektowej, itp.

W tej sytuacji wśród niektórych projektantów, w ślad za tym i wśród producentów sprzętu dla API, bywa głoszony obecnie pogląd, że jedyną potrzebą biur projektowych jest posiadanie sprzętu z bardzo podstawowym oprogramowaniem, natomiast wszelkie oprogramowanie użytkowe, użytkownicy sami sobie opracowują (o niesłuszności i społecznej szkodliwości tego poglądu - dalej).

Rozważania potrzeb sprzętowo-programowych, jakie w odniesieniu do API występują w biurach projektowych należy poprzedzić chociażby dość ogólnym usystematyzowaniem podstawowych pojęć w zakresie rodzajów prac projektowych i odpowiednim dla tych prac wyposażeniem sprzętowo-programowym.

Można przyjąć następującą klasyfikację prac projektowych, które w zależności od merytorycznej swojej treści mogą podlegać automatyzacji całkowicie, lub być w mniejszym, czy większym stopniu komputerowo wspomagane:

- prace obliczeniowe,
- wyszukiwanie informacji,
- generowanie dokumentacji.

Bardziej szczegółowe określenie form automatyzacji tych prac wymaga jeszcze dokładnych studiów i jest przedmiotem licznych i różnorodnych badań na całym świecie.

Można jednak wyodrębnić następujące sprzętowo-programowe wyposażenie wykorzystywane przy rea-

lizacji KWP.

- Oprogramowanie pomocnicze poziomu użytkowego, jak biblioteki numeryczne, techniczne, itp. i związane z nimi języki problemowe, a także systemy gospodarowania tym oprogramowaniem zapewniające prawidłową ich eksploatację.
- Zintegrowane systemy automatyzacji prac projektowych, czyli narzędzia programowe dla tworzenia konkretnych problemowo zorientowanych języków i odpowiadających im bibliotek podprogramów.
- Sprzętowe i programowe narzędzia dla tworzenia banków danych, czy to tzw. danych stałych dla danej dziedziny obiektów technicznych, czy też tworzenia tzw. numerycznych modeli danego obiektu projektowego. Oba zagadnienia różnią się dość znacznie i wymagają nieco odmiennych narzędzi programowych.
- W zakresie grafiki biernej: plottery i grafoskopy oraz ich oprogramowanie umożliwiające automatyczną generację dokumentacji projektowej końcowej lub automatyczną wizualizację wyników częściowych. Tego typu grafika jest niezbędnym elementem tych procesów projektowych, gdzie forma rysunku powstaje w wyniku automatycznego wykonania odpowiednich procedur projektowych (czy to obliczeniowych, czy też wyszukiwawczych), a jedynie forma graficzna umożliwia projektantowi ocenę prawidłowości realizacji zadania projektowego.
- W zakresie grafiki czynnej należy brać pod uwagę digitizery, monitory z piórem świetlnym, a obecnie i urządzenia rozpoznawania obrazu. Celowość stosowania tych urządzeń obserwuje się w tych procesach projektowych, gdzie pierwotne dane projektowe wyrażane są w formie graficznej. W tym przypadku zautomatyzowanie czynności przetwarzających formę graficzną do numerycznej niezbędnej dla dalszych przetwarzań, np. przez moduły obliczeniowe, pozwala wyeliminować jeden z najbardziej żmudnych elementów KWP - przygotowanie danych; a zatem obniżyć znacznie poziom błędów w danych. A właśnie wysoki poziom błędów danych w wielu przypadkach poważnie zmniejsza efektywność stosowania komputerów w projektowaniu. Ponadto urządzenia te wraz z odpowiednim oprogramowaniem pozwalają na realizację komputerowo wspomaganego procesu projektowego w formie interaktywnej. Jest to, jak się obecnie wydaje, forma najbardziej zbliżona do naturalnego, (a może tylko tradycyjnego) stylu pracy projektanta.

Osobno należy rozpatrywać takie procesy projektowe, gdzie już obecnie technologia wykonania zaprojektowanego wyrobu jest na tyle zautomatyzowana, że wyjście z komputerowego systemu projektującego dany obiekt powinno być wejściem do systemu sterującego wykonaniem tego obiektu.

Na przykład, z systemu projektującego wyrób mechaniczny, powinno być automatyczne połączenie do systemu generowania programów sterujących obrabianiem poszczególnych części (detali) tego wyrobu. Podobnie z systemu projektującego obwód drukowany do systemu sterującego wykonaniem płytek, itp.

Analiza potrzeb sprzętowych

Jak wynika z literatury, na podstawie specjalnych badań licznych organizacji projektowych w krajach wysoko rozwiniętych, ocenia się, że dla prawidłowej realizacji prac projektowych biura projektowe powinny być nasycone sprzętem różnej klasy. Przyczym generalnie wyróżnia się trzy po-

ziomy tego sprzętu:

- kalkulatory i kalkulatory programowalne,
- minikomputery pracujące w reżimie lokalnym,
- duże komputery zwykle, ale niekoniecznie, dostępne z inteligentnego terminalu, jakim jest lokalny minikomputer.

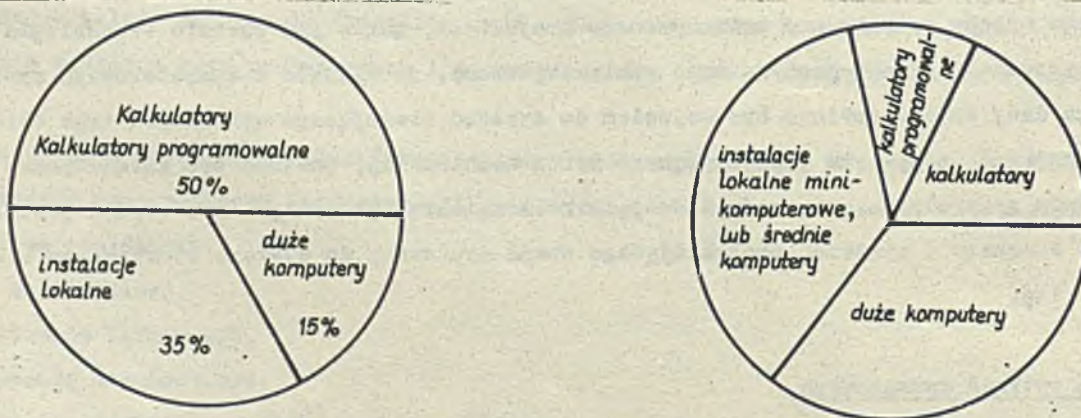
Do niedawna oceniano, że na kalkulatorach realizowano do 50% zadań projektowych, na instalacjach lokalnych około 35%, a na dużych komputerach, pozostałe 15%. Zakładano przy tym, że jeden kalkulator programowalny powinien obsłużyć od trzech do czterech projektantów.

Obecnie obserwuje się większe nasycenie organizacji projektowych systemami minikomputerowymi wyposażonych w 4 lub 6 końcówek konwersacyjnych, przy czym coraz częściej są to nie tylko dialogowe końcówki alfanumeryczne, ale i interaktywne końcówki graficzne.

Podobnie, jak w odniesieniu do kalkulatorów programowalnych, przyjmuje się, że z jednej końcówki powinno korzystać 3 lub 4 projektantów.

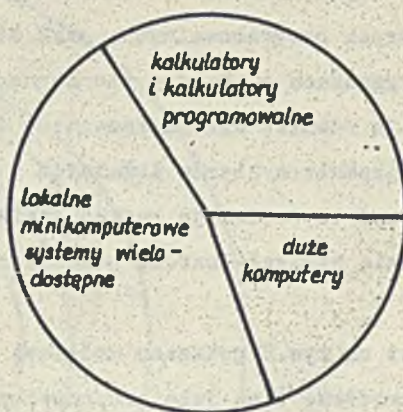
Warto podkreślić, że pomimo zdecydowanej tendencji do instalowania silnych instalacji lokalnych zdolnych rozwiązać większość zadań projektowych danego biura, zapewnia się możliwość bezpośredniego wejścia z instalacji lokalnej do sieci komputerowej (a nie jak dawniej do jednego centralnego komputera) w celu realizacji zadań projektowych bądź tak dużych, że moc instalacji lokalnej jest nie wystarczająca, bądź tak sporadycznie rozwiązywanych w danej organizacji projektowej, że nie opłaca się stale utrzymywać potrzebnego do tych zadań oprogramowania.

Można więc ocenić, jak kształtują się obecnie i jak powinny się kształtować, w perspektywie najbliższych 5 (8) lat, potrzeby wyposażenia biur projektowych w sprzęt konieczny dla prawidłowego rozwoju KWP. Przedstawiono to w formie graficznej, gdzie pokazano rozkład procentowy zautomatyzowanych prac projektowych pod względem wykorzystania trzech zasadniczych klas sprzętu API. Należy zwrócić uwagę, że na rysunkach tych uwzględnia się prace już zautomatyzowane, nie oznacza to jednak, zwłaszcza w odniesieniu do sytuacji krajowej, że obecnie zautomatyzowane są już wszystkie prace nadające się do automatyzacji. Należy tu również zauważyć, że specjalnie zaznaczono na rys.1b niski procent stosowania kalkulatorów programowalnych. Wyróżnienie to zostało dokonane dla zaznaczenia braków tego typu sprzętu w krajowej praktyce projektowej.



a) w wysoko rozwiniętych krajach 10(15) lat temu b) obecnie w kraju

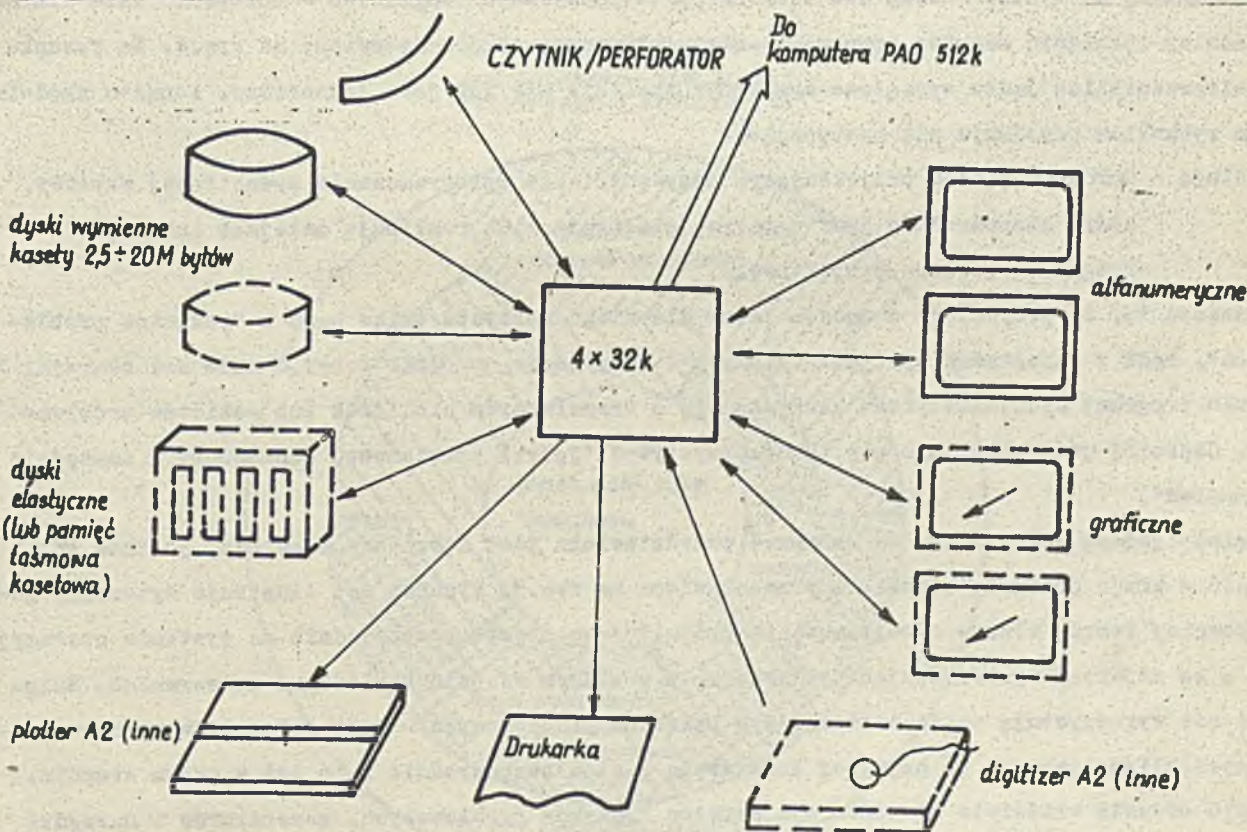
Rys.1. Struktura zautomatyzowanych prac inżynierskich (rys.1 c) i d) na stronie następnej)



c) obecna tendencja światowa

d) krajowa struktura docelowa w najbliższych 5 latach

Rys.1. Struktura zautomatyzowanych prac inżynierskich (rys.1 a) i b) na stronie poprzedniej)



Rys.2 Modułowa struktura lokalnego systemu minikomputerowego dla KWP

Na pozostałych rysunkach (1a, c, d) nie rozróżnia się prac wykonywanych na kalkulatorach i kalkulatorach programowalnych, gdyż nie ma to znaczenia z punktu widzenia metodyki projektowania. Na rysunkach a, b, d - nie zaznacza się połączenia systemów lokalnych z dużymi komputerami, gdyż to również nie ma znaczenia dla metodyki projektowania. Nie mniej na rys. 1c zaznaczono połączenie systemów lokalnych z siecią komputerową, a nie z pojedynczym dużym komputerem, gdyż jest to tendencja bardzo charakterystyczna. Oczywiście realizacja jej w przyszłości w kraju będzie wymagała bardzo poważnych prac sprzętowych i w zakresie oprogramowania podstawowego.

Natomiast na rys.2 pokazano modelową strukturę lokalnego systemu minikomputerowego. Nie należy interpretować tego jako proponowanej konfiguracji, ale jako ilustrację podstawowych potrzeb. Dlatego nie wymienia się na tym rysunku żadnych symboli, a jedynie te charakterystyki użytkowe, które decydują o przydatności danych urządzeń dla KWP.

Struktura oprogramowania projektującego

● Struktura oprogramowania systemów komputerowych

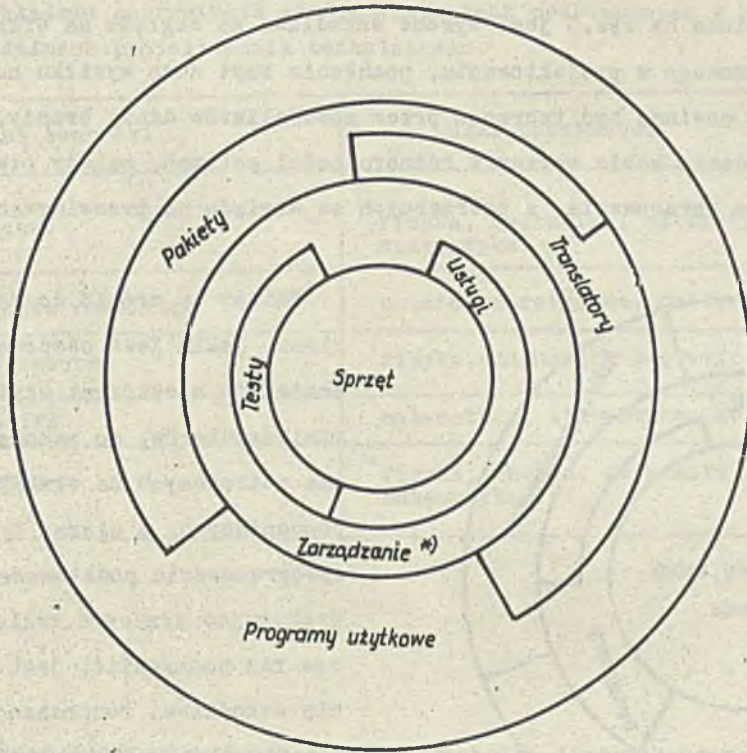
Przyjmując za punkt wyjścia klasyczny*) opis struktury oprogramowania systemów komputerowych przedstawiony na rys.3, należy dla ilustracji rozpatrywanych zagadnień - oprogramowania dla KWP dokładniej rozróżnić warstwy oprogramowania użytkowego, co przedstawiono na rys.4. Na rysunku tym nieprzenikliwe jądro sprzętowo-systemowe traktuje się już jako jednorodne. Ponadto zgodnie z tym rysunkiem przyjmuje się następującą:

Definicję - Oprogramowaniem projektującym nazywamy takie oprogramowanie zewnętrznej warstwy, które bezpośrednio jest pomocne, obsługuje, lub realizuje mniejsze lub większe fragmenty procesu projektowego.

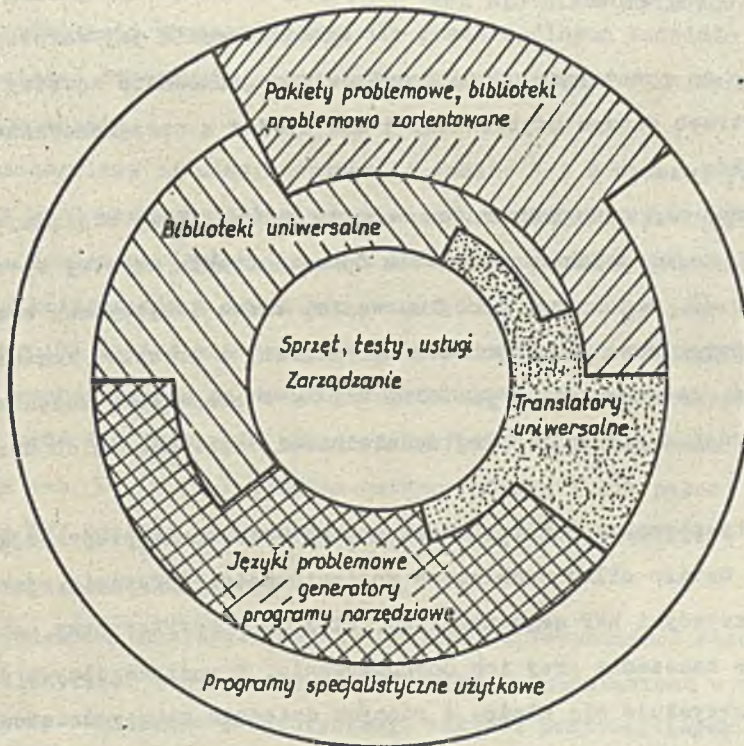
Oznacza to, że projektant w sposób jawny dla niego korzysta tylko bądź z "pakietów problemowych", bądź z "programów specjalistycznych". Oczywiście, zgodnie z rys.4 może się zdarzyć, że tworząc programy specjalistyczne korzysta się z translatorów bibliotek lub pakietów problemowych. Częściej jednak powinno się tu wykorzystywać "języki problemowe, generatory i narzędzia programowe".

Należy jednak podkreślić, że na rys.4 przedstawiona jest struktura docelowa, podczas gdy obecnie w kraju przeważa struktura przedstawiona na rys.5. Rysunek ten ilustruje sytuację, gdzie użytkownicy tworzą własne oprogramowanie projektujące często bezpośrednio na systemie operacyjnym, a co najwyżej wykorzystując translatory i w nikłym stopniu biblioteki uniwersalne. Natomiast nie wykorzystują wcale istniejących pakietów problemowych - jako składników oprogramowania specjalizowanego, a co najwyżej korzystają z nich bezpośrednio i to też w małym stopniu. Ponadto obecnie właściwie nie istnieje warstwa "języków problemowych, generatorów i narzędzi programowych".

*) Taka forma ujęcia struktury oprogramowania została zaczerpnięta z książki W.M.Turskiego "Propedeutyka informatyki"



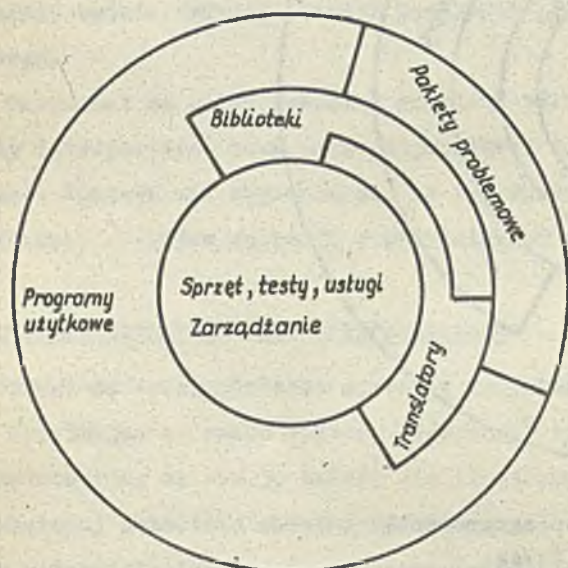
Rys. 3. Ogólna struktura oprogramowania
(na podstawie: W.M.Turski -
Propedeutyka informatyki)



Rys. 4. Docelowa struktura oprogramowania dla KWP

* Zarządzanie wraz z testami i usługami tworzy system operacyjny

Sytuacja przedstawiona na rys.5 jest wysoce szkodliwa ze względu na efektywność wykorzystywania sprzętu komputerowego w projektowaniu, pochłania zbyt dużo wysiłku na tworzenie warstwy użytkowej. Warstwa ta powinna być tworzona przez specjalistów danej branży, czy dziedziny techniki, przy czym dla zdania sobie sprawy z różnorodności potrzeb, należy przeanalizować rodzaje programów możliwych do opracowania, a potrzebnych ze względu na przewidywaną powszechność zastosowania.



Rys.5. Obecne struktury oprogramowania dla KWP

dzy użytkownikami, jako że powstające u użytkowników oprogramowanie warstwy narzędziowej jest zbyt dostosowane do potrzeb danego użytkownika i nie spójne z oprogramowaniem opracowywanym przez innych użytkowników.

Z drugiej strony prace nad oprogramowaniem warstwy narzędziowej odciągają zespoły programistyczne w organizacjach projektowych od tworzenia oprogramowania warstwy specjalizowanych programów użytkowych (por.rys.4), czyli tego oprogramowania, które dotyczyłoby wysoce wyspecjalizowanych problemów projektowych rozwiązywanych w danym biurze. A przecież właśnie to oprogramowanie dopiero może spowodować, że włączenie komputerów w proces projektowy zacznie przynosić efekty; odwołując się do chwili podważa się sens całej działalności zmierzającej do wprowadzenia metod KWP.

• Klasyfikacja potrzeb

Jedną z charakterystycznych cech bezpośredniego oprogramowania projektującego (zewnętrzna warstwa na rys.4) jest bardzo silne problemowe zorientowanie. Wprawdzie, jak wykaże się dalej, nie jest to cechą dla metodyki KWP najważniejszą, niemniej merytoryczny zakres poszczególnych programów ma zasadnicze znaczenie przy ich opracowywaniu. W poszczególnych dziedzinach techniki przy projektowaniu wykorzystuje się wiedzę z różnych dziedzin nauk podstawowych, co przykładowo ilustruje tablica 1. Zazwyczaj treści zaczerpnięte z nauk podstawowych przy KWP przybierają formę modułów obliczeniowych. Nie wyczerpuje to możliwych form komputerowego wspomagania projektowego, a zwłaszcza możliwych form automatyzacji poszczególnych prac projektowych. W zależności od konkretnych poszczególnych procesów projektowych znajdują się w nich zastosowania różnorodne

Należy tu wrócić do sygnalizowanego problemu, jakim jest obserwowana obecnie w kraju tendencja niektórych użytkowników, zwłaszcza minikomputerów, do samodzielnego opracowywania potrzebnych na własny użytek bibliotek pomocniczych, a nieraz i pewnych elementów oprogramowania podstawowego (por. rys.5). Nie dyskutując przyczyn takiego stanu, warto jeszcze raz podkreślić: jest to tendencja społecznie szkodliwa, rozprasza siły, doprowadzając do wielokrotnego powtarzania niektórych prac. Opracowywanie oprogramowania warstwy narzędziowej przez poszczególnych użytkowników nie gwarantuje bynajmniej warunków do wymiany oprogramowania tej warstwy bezpośrednio pomiędzy

Tablica 1. Przykładowe zestawienie zastosowania nauk podstawowych w wybranych dziedzinach projektowania technicznego

Lp.	Dziedzina techniki	Nauki podstawowe
1	Budownictwo	fizyka, mechanika, matematyka, chemia, statystyka
2	Technologia chemiczna	chemia teoretyczna, matematyka, fizyka
3	Maszynoznawstwo	fizyka, mechanika teoretyczna, matematyka
4	Elektronika	matematyka, statystyka, kombinatoryka
5	Metalurgia	fizyka, chemia, mechanika teoretyczna, matematyka

rodzaje programów.

W tablicy 2 zestawiono wykaz niektórych dziedzin techniki i przykładowych procesów projektowych ze szczególnie trudnymi lub żmudnymi pracami tych procesów i przykładowymi rodzajami programów, jakie mogą zautomatyzować te prace.

Należy podkreślić, że obie te tablice jedynie ilustrują problem, rzeczowej klasyfikacji potrzeb w zakresie oprogramowania projektującego dla poszczególnych dziedzin techniki, nie rozwiązując go jednak całkowicie. Stworzenie pełnej klasyfikacji rzeczowej wymaga dużo szerszych i bardziej pogłębionych analiz.

Ponadto, co jest szczególnie istotne, porównanie tablic 1 i 2 wykazuje, że merytoryczno-rzeczowa analiza procesów projektowych jest niewystarczająca dla identyfikacji potrzeb programistycznych na rzecz KWP. Konieczna jest w tym celu analiza sposobu realizacji procesu projektowego, uwzględniająca sposoby korzystania z komputera, czyli typy i zakresy prac projektowych automatyzalnych. Wstępne wyniki takiej analizy przedstawia tablica 3. Uwidoczniono w niej, z których warstw oprogramowania rozróżnianych na rys.4 korzysta się bądź bezpośrednio w poszczególnych pracach projektowych, bądź dla tworzenia ostatecznej formy oprogramowania projektującego.

Należy zauważyć, że tab. 3 ujmuje problem bardzo ogólnie i dla pełnej identyfikacji rodzajów oprogramowania dla KWP powinny być prowadzone dalsze bardziej wnikliwe analizy.

• Analiza rodzajów oprogramowania dla KWP

Opierając się na wymienionych klasyfikacjach oraz na zaproponowanej strukturze oprogramowania można przyjąć ogólną klasyfikację oprogramowania dla KWP, przedstawioną w tab.4.

Ostateczne szczegółowe określenie merytorycznego zakresu poszczególnych typów i klas oprogramowania wymaga dodatkowego przeanalizowania, które klasy zastosowań przeważają, a to w celu wybrania takich typów programów niższych warstw, np. translatorów, zakresów bibliotek, generatorów, itp., które znajdują najbardziej powszechne zastosowanie.

Tyb. 2. Alfabetyczne zestawienie dziedzin techniki procesów projektowych i komputeryzowalnych prac projektowych

Lp.	Dziedzina	Wybrany proces projektowy	Przykłady prac projektowych		Rodzaj oprogramowania
			szczególnie trudna	szczególnie żmudna	
1	Architektura	forma budowli	konopowanie kształtu	-	grafika interaktywna
2	Automatyka	projekt instalacji	-	zestawienie całości z projektowanymi elementami	generowanie dokumentacji na podstawie katalogu
3	Budownictwo	projekt konstrukcyjny	analiza wytrzymałościowa	-	obliczenia numeryczne
4	Chemia	proces technologiczny	-	robocze dokumentowanie postępu prac projektowych	numeryczny model obiektu
5	Drogownictwo	projekt linii komunikacyjnych	analiza reakcji chemicznych	-	obliczenia numeryczne
6	Elektrotechnika	projekt instalacji	-	przetworzenie danych z pomiarów oraz/lub z podziałów geodezyjnych	automatyzacja wprowadzania danych a) bezpośrednio z pomiarów b) z rysunku
			określenie parametrów tych elementów	gospodarowanie wielką liczbą danych	numeryczny model obiektu

Tab. 3. Formalna klasyfikacja komputeryzowanych prac projektowych

Sposoby korzystania z komputera								
Sposób realizacji procesu projektowego	Obliczenia	Automatyzacja wprowadzenia danych		Wyszukiwanie informacji stałych	Tworzenie numerycznego modelu obiektu	Wizualizacja wyników pośrednich		Automatyzacja dokumentacji
		alfanumerycznych	digitalizacja			trwała	na monitorze ekranowym	
Projektowanie oalkowicie nowych rozwiązań konstrukcyjnych wg nowych metod nowych modeli	Nowe metody, nowe algorytmy, oprogramowanie powstaje z użyciem translatora ewentualnie biblioteki uniwersalnej	Pisanie własnych fragmentów ewentualnie z wykorzystaniem generatorów	Korzystanie z programów narzędziowych	Korzystanie z programów narzędziowych	Korzystanie z generatorów i programów narzędziowych	Pisanie własnych fragmentów ewentualnym wykorzystaniem generatorów	wydruki rysunki	
Nowe konstrukcje wg znanych metod	Wykorzystanie bibliotek uniwersalnych języków problemowych i ewentualnie bibliotek problemowych	Pisanie własnych fragmentów ewentualnie z wykorzystaniem generatorów	Korzystanie z języków problemowych i programów narzędziowych	Korzystanie z języków problemowych i programów narzędziowych	Korzystanie z bibliotek z języków uniwersalnych i programów narzędziowych	Korzystanie z generatorów i programów narzędziowych		
Nowe konstrukcje ze znanych elementów, których parametry wylicza się i dobiera	Korzystanie z języków problemowych oraz pakietów i bibliotek problemowych	Korzystanie z generatorów	Korzystanie z pakietów problemowych	Korzystanie z pakietów problemowych	Korzystanie z pakietów i bibliotek problemowych oraz z generatorów i programów narzędziowych	Korzystanie z pakietów problemowych i bibliotek problemowych oraz z generatorów i programów narzędziowych		
Kompletowanie całej konstrukcji z elementów typowych katalogowych	Korzystanie z języków problemowych lub pakietów problemowych	Korzystanie z pakietów problemowych i generatorów	Korzystanie z pakietów problemowych	Korzystanie z pakietów problemowych	Korzystanie z pakietów problemowych lub języków problemowych i generatorów	Korzystanie z pakietów problemowych, generatorów i programów narzędziowych		

Tab. 4. Rodzaje oprogramowań w poszczególnych warstwach

		Komputeryzowalne czynności projektowe							
Warstwa	Obliczenia	Automatyzacja wprowadzania danych		Wyszukiwanie informacji stałych	Tworzenie numerycznego modelu obiektu	Wizualizacja wyników posrednich		Automatyzacja dokumentacji	
		alfanumerycznych	digitalizacja			trwała	na monitorze ekranowym	wydruki	rysunki
Translatory uniwersalne	Języki algorytmiczne	Języki konwersyjne	Oprogramowanie bazowe urzędzenia	Języki gospodarki ram	Oprogramowanie bazowe urzędzeń			Oprogramowanie bazowe urzędzeń	
Biblioteki uniwersalne	Biblioteki numeryczne i statystyczne	Procedury uniwersalne	Biblioteki urzędzeń	Uniwersalne procedury				Biblioteki urzędzeń i uniwersalne procedury	
Języki problemowe, generatory i programy narzędziowe	Języki dla poszczególnych dziedzin	Generatory dialogu i kontroli danych	Generatory digitalizacji	Programy narzędziowe dla gospodarki zbiorami	Generatory rysunków i wydruków			Generatory wydruków, języki re-dakcji tekstów	Generatory rysunków
Pakiety problemowe i biblioteki	Programy i podprogramy silnie zorientowane	Programy i procedury czytania określonych klas danych	Programy i procedury digitalizacji określonych klas rysunków	Systemy bankowe stałych danych projektowych dla określonych obiektów technicznych	Programy i podprogramy dla określonych klas tworzenia obrazów i testów w trybie interaktywnym			Problemowe programy redakcyjne rysujące	
Wyspecjalizowane programy użytkowe		Programy i fragmenty programów wąsko wyspecjalizowane							

Wnioski

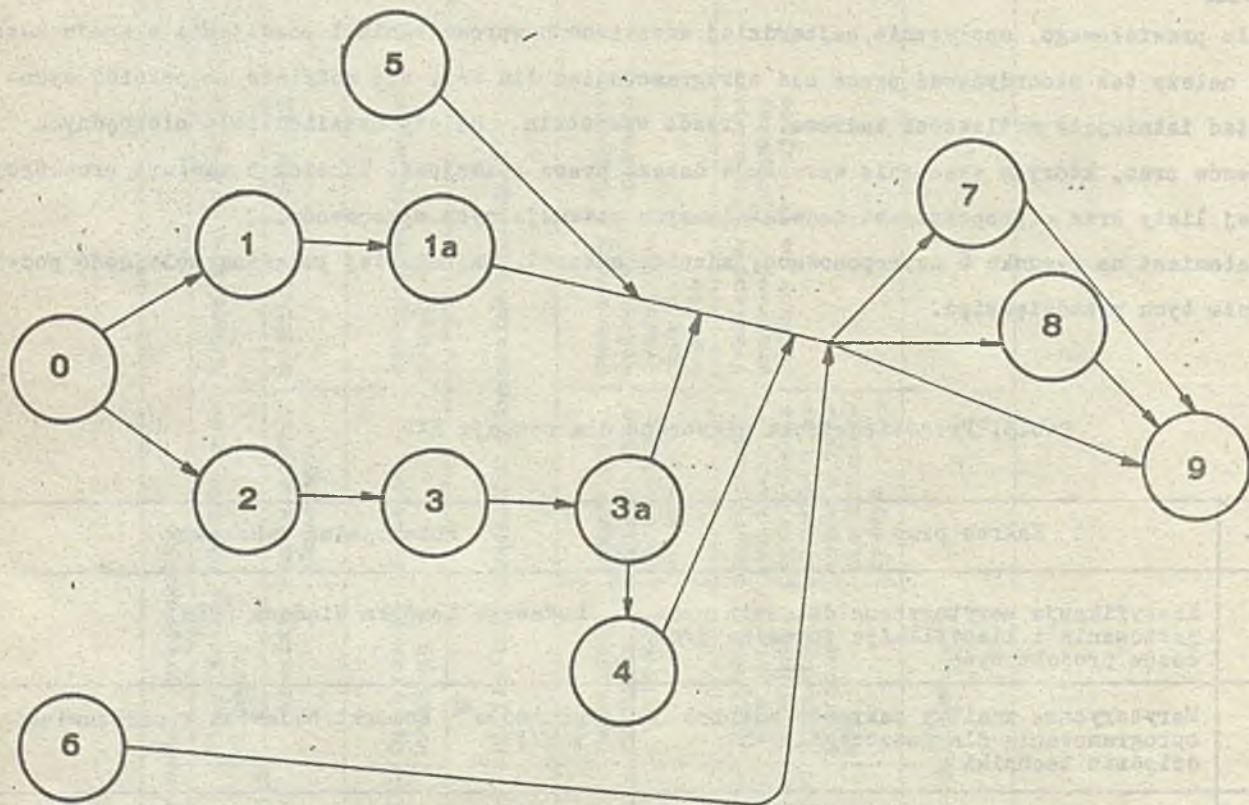
Dla prawidłowego, społecznie, najbardziej efektywnego wprowadzania i rozwijania w kraju metod API, należy tak skoordynować prace nad oprogramowaniem dla KWP, aby możliwie najpełniej wykorzystać istniejące możliwości kadrowe. Przede wszystkim należy ustalić listę niezbędnych zakresów prac, których wykonanie warunkuje dalsze prace rozwojowe. Tablica 5 zawiera propozycje takiej listy wraz z propozycjami najważniejszych potencjalnych wykonawców.

Natomiast na rysunku 6 zaproponowano, zdaniem autorki, najbardziej pożądaną kolejność podejmowania tych przedsięwzięć.

Tab.5. Przedsięwzięcia niezbędne dla rozwoju KWP

Lp.	Zakres prac	Potencjalni wykonawcy
0	Klasyfikacja merytoryczna dziedzin projektowania i klasyfikacja formalna procesów projektowych	Badawcza Komórka Wiodąca (BKW)
1	Merytoryczne analizy zakresów potrzeb oprogramowania dla poszczególnych dziedzin techniki	Branżowe ^{*)} komórki badawcze w porozumieniu z BKW
1a	Synteza tych potrzeb	BKW w uzgodnieniu z komórkami branżowymi
2	Formalne analizy przebiegów różnych procesów projektowych	Branżowe komórki metodyczne we współpracy z BKW
3	Analizy formalnych potrzeb tych procesów w zakresie API	Komórki branżowe we współpracy z BKW
3a	Synteza tych potrzeb	BKW w uzgodnieniu z komórkami branżowymi
4	Analiza możliwości i potrzeb wyposażenia sprzętowego	Inne komórki naukowe Zjednoczenia we współpracy z BKW
5	Analiza stanu i tendencji światowych w zakresie KWP	BKW we współpracy z innymi krajowymi placówkami badawczymi
6	Prace nad oprogramowaniem bazowym urządzeń, systemami operacyjnymi, translatorami, bibliotekami uniwersalnymi	Odpowiednie działy programistyczne Zjednoczenia z wiedzą BKW
7	Prace nad językami problemowymi, generatorami i oprogramowaniem narzędziowym	Z inicjatywy BKW zlecane specjalistycznym działom w Zjednoczeniu lub innym instytucjom informatycznym
8	Prace nad pakietami problemowymi i bibliotekami problemowo zorientowanymi	Z wiedzą BKW prowadzone przez właściwe placówki programistyczne
9	Prace nad wyspecjalizowanym oprogramowaniem użytkowym	Realizowane przez dowolnych użytkowników

^{*)} Branżowe oznacza tu: resortowe, uczelniane lub PAN-owskie komórki badawcze odpowiednie dla danej dziedziny techniki



Rys.6. Kolejność przedsięwzięć w zakresie rozwoju oprogramowania dla KWP

Uwaga: Znaczenie numerów jak w tab.5.

mgr inż. Jerzy MOCAŁA
Instytut Maszyn Matematycznych

Wprowadzenie do systemu MUMPS-11

Stosowanie maszyn cyfrowych w procesie projektowania napotyka na liczne bariery. Jedną z nich wg O. Bartkiewicz-Stulińskiej [1] jest niechęć projektantów do uczenia się nowych technik, a w szczególności uczenia się obsługi i działania maszyn cyfrowych. Sytuacja poprawiła się, gdy na rynku komputerowym pojawiły się małe maszyny (WANG, NOVA) z językiem programowania BASIC.

Prostota obsługi systemu przybliżyła projektantów do klawiatury terminala maszyny cyfrowej. Bezpośredni kontakt projektanta z maszyną umożliwił wykorzystanie w procesie projektowania systemów informacyjnych (baz danych) np. systemu norm, katalogów elementów, itp.

Przez analogię do rozwoju techniki baz danych, która wykazała, że racjonalne wykorzystanie tej techniki jest możliwe w nowym typie maszyn można sądzić, że dalszy rozwój stosowania komputerów w procesie projektowania należy oprzeć na systemie klasy BASIC zmodyfikowanym o mechanizm umożliwiający łatwe zakładanie zbiorów i przetwarzanie informacji magazynowanej w zbiorach.

Takim systemem jest system MUMPS-11 (Massachusetts General Hospital Utility Multi - Programing System) implementowany na maszynie PDP-11.

Wprawdzie system ten został zaprojektowany dla potrzeb służby zdrowia, ale ma wiele cech, którymi powinny charakteryzować się systemy do prac projektowych.

Ponieważ system MUMPS jest implementowany na maszynie MERA-400 (jako ostatnia warstwa systemu operacyjnego SOM-5) celowe jest bliższe z nim zaznajomienie.

Opisując dyrektywy i instrukcje języka przyjęto następujące zasady:

- parametry w nawiasach trójkątnych < > muszą być zastąpione wartościami podanymi przez użytkownika,
- nawiasy kwadratowe [] oznaczają, że parametry w nich zawarte są opcjonalne,
- prostokąty oznaczają klawisze zwrotnego urządzenia wejścia/wyjścia, np. CR - "powrót karetki".

Przedstawiając system MUMPS przyjęto założenie, że system cyfrowy składa się z procesora, pamięci operacyjnej, dysku, zwrotnego urządzenia wejścia/wyjścia (jest to typowy zestaw systemu MERA-400).

Ogólna charakterystyka systemu MUMPS

MUMPS jest systemem konwersacyjnym, wielodostępnym, umożliwiającym użytkownikowi pracę w trybie natychmiastowym i w trybie programowym. Jest bardzo podobny do systemu BASIC [2].

W trybie natychmiastowym wykonuje się dyrektywy typu: wyświetlenie wartości zmiennych, czyszczenie pamięci operacyjnej, krótkie obliczenia, przekazanie programu do wykonania.

W trybie programowym wykonuje się przygotowany program w algorytmicznym języku programowania.

Program wprowadza się do pamięci maszyny wiersz po wierszu. Każdy wiersz programu opatrzony jest numerem, który pozwala ustalić kolejność wykonywania wierszy oraz odróżnić wiersz programu od dyrektywy wykonywanej w trybie natychmiastowym.

Numer wiersza, nazywany numerem kroku jest liczbą dziesiętną dodatnią z przedziału 0.01 - 327.67. Numer nie może mieć zerowej części ułamkowej, np. liczby 1.00, 20.00 nie są numerami kroków.

Wiersze o takich samych częściach całkowitych numerów tworzą logiczne segmenty programu będące odpowiednikami podprogramów w innych językach algorytmicznych.

W systemie MUMPS, podobnie jak w systemie BASIC, tryby natychmiastowy i programowy mogą się przeplatać, a także część dyrektyw systemowych ma swoje odpowiedniki w instrukcjach programowych.

Tryb natychmiastowy. Dyrektywy systemu MUMPS

Gotowość przyjęcia zlecenia przez system sygnalizowana jest wydrukiem znaku >. Po tym znaku można napisać dyrektywę zlecającą wykonanie zamierzonej pracy. W tab.1 podano podstawowy zestaw dyrektyw MUMPS z odpowiednikami systemu BASIC [2].

Tab.1. Dyrektywy systemu MUMPS

Lp.	Dyrektywa	Interpretacja	Odpowiednik w BASIC-u
1	ERASE]	Wyczyszczenie pamięci operacyjnej, wymazanie programu z PAO	CLEAR
2	WRITE]	Wydruk programu zmagazynowanego w pamięci operacyjnej	LIST
3	HALT]	Zakończenie sesji	-
4	DO]	Wykonanie programu, przejście do trybu programowego	RUN
5	FILE]	Przesłanie programu do pamięci dyskowej	SAVE
6	LOAD]	Ładowanie programu z pamięci dyskowej do PAO	LOAD
7	TYPE]	Drukowanie wartości zmiennych	PRINT
8	KILL]	Wymazywanie zmiennych	CLEAR V
9	SET]	Kreowanie zmiennych	LET
10	MODIFY]	Modyfikacja wiersza programu	-

System nie wymaga pisania całego słowa dyrektywy. Wystarczy napisać tylko pierwszą literę dyrektywy.

Przykład
Dyrektywa postaci > E jest równoważna z dyrektywą > ERASE (wyczyszczenie pamięci operacyjnej).

Oprócz tych dyrektyw, użytkownik systemu MUMPS dysponuje dyrektywami umożliwiającymi poprawę popełnionych błędów. Zbiór takich dyrektyw przedstawiono w tab.2.

Tab.2. Dyrektywy pomocnicze systemu MUMPS

Lp.	Dyrektywa	Interpretacja
1	RUB OUT	Wymazanie pojedynczych znaków
2	CTRL i U	Wymazanie wiersza
3	CTRL i C	Zatrzymanie wykonywania programu - przejście do trybu natychmiastowego

Przykład
Użytkownik po zgłoszeniu się systemowi (znak ">") chciał wyczyścić pamięć operacyjną (dyrektywa ERASE). Napisał dyrektywę postaci:

> ERASEE

Przed przekazaniem dyrektywy do wykonania (klawisz **CR**) spostrzegł błąd (zbędna litera E). Naciśnięcie klawisza **RUB**
OUT może wymazać niepotrzebną literę. Każde naciśnięcie tego klawisza sygnalizowane jest wydrukiem znaku "/": > ERASEE **RUB**
OUT spowoduje usunięcie ostatniego znaku. Wydruk wiersza będzie postaci: > ERASEE/ .

Zamiast wymazywać niepotrzebny znak można wymazać cały wiersz i napisać poprawnie dyrektywę. Naciśnięcie klawiszy **CTRL** i **U** spowoduje się wymazanie całego wiersza:

> ERASEE **CTRL** i **U**

Na wydruku wiersz taki zostanie opatrzone znakami "↑U":

> ERASEE ↑U

System po wykonaniu dyrektywy "wymazanie wiersza" zgłosi gotowość przyjęcia nowej dyrektywy (znak ">").

Poniżej zamieszczono krótki opis dyrektyw z tab.1.

● Czyszczenie pamięci operacyjnej

Postać ogólna dyrektywy jest następująca

E[RASE][< n₁ >, < n₂ >,]

Gdzie: n₁, n₂,... są numerami kroków (wierszy).

Za pomocą dyrektywy wymazuje się z pamięci operacyjnej wyspecyfikowane kroki przechowywanego programu.

Przykłady

```
>ERASE (wymazanie całego programu z PAO)
>E 1.38 (wymazanie kroku o numerze 1.38)
>E 1.4.1.5.2.6 (wymazanie trzech kroków o numerach 1.4,1.5,2.6)
```

● Wydruk programu

Program w języku MUMPS przechowywany jest w pamięci operacyjnej. Żeby otrzymać listing programu używa się dyrektywy postaci:

```
W[RITE] [<n1> , <n2> , ...]
```

gdzie n_1, n_2, \dots są numerami kroków.

Dyrektywa bez parametrów spowoduje wyprowadzenie na urządzenie wyjściowe całego programu zmagazynowanego w PAO. W wypadku użycia dyrektywy z parametrami drukowane są jedynie wyspecyfikowane kroki programu.

Przykłady

```
>W (wydruk całego programu)
>W 1.1, 1.2 (wydruk kroku 1.1 i 1.2)
```

● Zakończenie sesji

System MUMPS obsługuje wielu użytkowników. Odłączenie się użytkownika od systemu (zakończenie pracy użytkownika) następuje za pomocą dyrektywy H[ALT].

System po wykonaniu dyrektywy drukuje komunikat: EXIT.

● Wykonanie programu

Przejdźcie do trybu programowego, w którym wykonywany jest program znajdujący się w PAO realizuje się dyrektywą:

```
D[0] <n1> [<n2> , <n3> , ...]
```

gdzie n_1, n_2, n_3 - numery kroków programu.

Przykłady

```
>D 1.1 Wykonanie kroku o numerze 1.1
>D 1 wykonanie wszystkich kroków, których część całkowita numeru równa się 1
>D 1.1,2,3.8 Wykonanie kroku 1.1, części 2 i kroku 3.8.
```

● Przesłanie programu do pamięci dyskowej

Program rezydujący w pamięci operacyjnej może być zapamiętany na dysku w wyniku wykonania dyrektywy:

F[ILE] <nazwa₁> [, <nazwa₂> , <nazwa₃> ...]

gdzie nazwa₁, nazwa₂, nazwa₃ - nazwy kopii programu na dysku.

Przykłady

- >F PGM Program z pamięci operacyjnej zostanie przesłany na dysk i zapamiętany w obszarze o nazwie PGM
- >F PGM1, PGM 2 Program z PAO zostanie przesłany do dwóch miejsc na dysku o nazwach PGM1, PGM2.

• Ładowanie programu do pamięci operacyjnej

Program zapamiętany na dysku może być ściągnięty do PAO za pomocą dyrektywy postaci:

I[OAD] < nazwa >

gdzie nazwa - przyjęta nazwa w dyrektywie FILE.

Przykłady

- >LOAD CZYT Do pamięci ma być załadowany program o nazwie CZYT. Gdy nie ma takiego programu, system wydrukuję komunikat
- NOPGM> Ø @

• Drukowanie wartości zmiennych

Dyrektywa drukowania ma postać:

T [YPE] < lista >

gdzie lista - ciąg zmiennych, stałych, wyrażeń arytmetycznych i znaku zmiany wiersza "!".

Przykład

Przyjmując, że zmienna $A=0.05$, $B=10.0$ dyrektywa:

>TYPE "A=",A," ", "B=",!, "A*B=",A*B

wygeneruje wydruk:

$A=0.05$ $B=10$

$A*B=0.5$

Dyrektywa postaci:

>T "A=",A,"B=",B,"A*B=",A*B

wygeneruje wydruk:

$A=0.05B=10A*B=0.5$

• Kreowanie zmiennych

W systemie MUMPS rozdzielono zmienne od operacji wykonywanych na tych zmiennych zapisanych w postaci programu. Dyrektywa kreowania zmiennych wprowadza identyfikator zmiennej do słownika zmiennych i ustala wartość zmiennej.

Postać ogólna dyrektywy jest następująca:

S[ET] < v₁ > = < w₁ > [, < v₂ > = < w₂ > , ...]

gdzie: $v_1, v_2 \dots$ - identyfikator zmiennych,
 $w_1, w_2 \dots$ - wartość zmiennych; wartościami zmiennych może być stała, identyfikator
wykreowanej zmiennej, wyrażenie.

Przykład

>S A=1, B="STAŁA"

>S C=A*B

● Wymazywanie zmiennych

Wymazywanie zmiennych wykreowanych w systemie dokonuje się za pomocą dyrektywy postaci:

$$K[ILL][\langle v_1 \rangle, \langle v_2 \rangle, \dots]$$

gdzie: v_1, v_2, \dots - identyfikatory zmiennych.

Dyrektywa usuwa wyspecyfikowane zmienne ze słownika zmiennych. W wypadku użycia dyrektywy bez parametrów, ze słownika będą wymazane wszystkie zmienne wykreowane w systemie.

Przykład:

>K A,B (wymazanie zmiennych A i B)

>K (wymazanie wszystkich zmiennych)

● Modyfikacja wiersza programu

Wprowadzony do pamięci operacyjnej wiersz programu można poprawić używając dyrektywy postaci:

$$M[ODIFY] \langle nr \rangle : \langle d \rangle \langle \text{łańcuch 1} \rangle \langle d \rangle \langle \text{łańcuch 2} \rangle$$

gdzie: nr - numer kodu

d - (delimiter) dowolny znak, który nie występuje w łańcuchu 1 i 2

łańcuch 1 - łańcuch znaków, który chcemy poprawić,

łańcuch 2 - łańcuch znaków, który ma być wpisany w miejsce łańcucha 1.

Przykład

W wierszu programu:

```
1.1Ø SET A=A*B/D
```

dzielnik D chcemy zamienić na wyrażenie A+C.

Używając dyrektywy postaci:

```
>M 1.1:* D*A+C
```

otrzymamy:

```
1.1Ø SET A=A*B/A+C
```

Na zakończenie tego krótkiego przeglądu dyrektyw trybu natychmiastowego systemu MUMPS warto zwrócić uwagę na możliwość pisania programów wykorzystujących powyższe dyrektywy. W jednym wierszu można umieścić ciąg dyrektyw w kolejności ich wykonywania. Wypisane dyrektywy będą realizowane automatycznie.

Przykład

Poniższe dyrektywy:

```
>E          Czyszczenie pamięci
>L PGM      Załadowanie programu z dysku
>D 4        Wykonanie programu
>T A,I,B,I,"KONIEC"  Wydruk (np. kontrolny) wartości zmiennych można wykonać następująco:
>E L PGM D 4 T A,I,B,I,"KONIEC"
```

Tryb programowy. Opis języka MUMPS

W trybie programowym wykonuje się programy napisane w języku MUMPS. Program zbudowany jest z wierszy programu zawierających instrukcje. Każdy wiersz opatrzony jest numerem, nazywanym numerem kroku. Tworzenie wierszy programu odbywa się w trybie natychmiastowym. Przed wprowadzeniem wiersza do PAO można go poprawić używając dyrektyw z tab.2.

Wiersz zapamiętany w pamięci operacyjnej poprawia się dyrektywą MODIFY (tab.1).

Przejsięcie z trybu natychmiastowego do programowego realizuje się za pomocą dyrektywy DO.

Przejsięcie z trybu programowego do natychmiastowego dokonuje się automatycznie po zakończeniu wykonywania się programu lub w sytuacjach awaryjnych (p. zapętlenie się programu) za pomocą dyrektywy **CTRL** i **C** :

Poniżej przedstawiono elementy języka MUMPS.

• Alfabet języka

Litery: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cyfry: 0123456789

Symbole: !"#\$%&'() * = < @ [\ + ^] < > ? : - ; . /

Typy stałych

W programach można używać stałych liczbowych i alfanumerycznych. Stałe liczbowe muszą być liczbą z przedziału ± 21474736.74 . Niedopuszczalne jest używanie liczb o większej liczbie cyfr znaczących po kropce dziesiętnej niż dwie. Stałe alfanumeryczne nie mogą być ciągiem znaków zawierającym więcej niż 132 znaki.

Przykład

1., -4378.54 - stałe liczbowe

"DOWOLNE ZNAKI" - stała tekstowa

W przypadku użycia w programie niedozwolonych stałych drukowane są odpowiednie komunikaty.

Przykład

Wiersz programu ma postać:

1.1 T A + 0.045

System wydrukuje komunikat:

MINIM > 1.10.@

Gdyby stała była następująca: 40 000 999, komunikat miałby postać:

MXNUM > 1.10 (d)

● Zmienne

Zmienna jest nazwą mającą w systemie wartość, która w trakcie wykonywania programu może się zmieniać. Przyjmując różne kryteria można dokonać różnej klasyfikacji zmiennych. W tab.3 przedstawiono możliwie najpełniejszy podział zmiennych umożliwiający opis języka MUMPS.

Tab.3. Klasyfikacja zmiennych

Lp.	Kryterium	Typy zmiennych
1	Wartość	liczbowa, alfanumeryczna
2	Struktura	prosta, indeksowana
3	Miejsce przechowywania wartości	lokalna (pamięć operacyjna), globalna (dysk)
4	Wykonywane operacje	programowe, systemowe

Podział zmiennych ze względu na wartość jest mało istotny ze względu na to, że w programie nie deklaruje się zmiennych (por. instrukcja DIM w BASIC), a pod tym samym identyfikatorem można przechowywać wartość liczbową lub alfanumeryczną, byleby liczba nie przekraczała ± 21474836.47 , a łańcuch

alfanumeryczny nie miał więcej niż 132 znaki.

Zmienna indeksowana jest odpowiednikiem tablioy w innych językach programowania. Indeks może być stała, zmienna lub wyrażenie arytmetyczne. Wartością indeksu jest liczba rzeczywista dodatnia. Zakres wartości indeksu jest ograniczony i wynosi $\emptyset - 20975.51$. Zmienna indeksowana przechowywana w PAO może mieć tylko jeden indeks (tablica jednowymiarowa), natomiast zmienna przechowywana na dysku może być dwuindeksowa (tablica dwuwymiarowa).

Najważniejszym podziałem zmiennych jest klasyfikacja ze względu na miejsce przechowywania wartości. Zmienne, których identyfikatory rozpoczynają się od znaku "↑" (por. tab.4) lokowane są na dysku. Zmienne te nazywane globalnymi ułatwiają komunikację między programami, a także w istotny sposób upraszczają zakładanie plików. Mechanizm zmiennych globalnych będzie zilustrowany w dalszej części artykułu przykładem programu zakładającego zbiór informacji zawartych w Polskich Normach i wyszukującego potrzebne projektantowi informacje.

Przykłady identyfikatorów omawianych zmiennych przedstawiono w tab.4.

Tab.4. Przykłady identyfikatorów zmiennych

Typ zmiennych	Zmienna lokalna prosta	Zmienna globalna prosta	Zmienna lokalna indeksowana	Zmienna globalna indeksowana
Identyfikator	A, B12 B1,ABC	↑A, ↑B12 ↑B1, ↑ABC	A(X), B12(5/2) B1(A(X)), ABC(1)	↑A(X), ↑ABC(1), ↑B1(↑A(X), X)

Na zmiennych można wykonywać działania przedstawione w tab.5.

Tab.5. Działania w systemie MUMPS

Nazwa działania	Symbol	Typ wartości
dodawanie	+	liczbowe, znakowe
odejmowanie	-	liczbowe, znakowe
mnożenie	*	liczbowe, znakowe
dzielenie	/	liczbowe, znakowe
konkatenacja	@	znakowe

Blizszego wyjaśnienia wymagają jedynie działania arytmetyczne wykonywane na wartościach będących łańcuchami znaków.

Działania takie wykonuje się wg zasady:

- przeszukuje się łańcuch znaków w celu znalezienia pierwszego podciągu zawierającego cyfry i kropkę dziesiętną,

- przekształca się te podciągi na wew-

nętrzną reprezentację liczb w systemie i wykonuje działanie arytmetyczne na liczbach.

Przykład

Następujące działanie:

"OBJETOSC: 84"/"POLE PODSTAWY 4"

da wynik: 21

- Instrukcja podstawienia

Instrukcja ustala wartość zmiennej (por. dyrektywa SET). Postać ogólna instrukcji:

$$S[ET] \langle v_1 \rangle = \langle w_1 \rangle [\langle v_2 \rangle = \langle w_2 \rangle , \dots]$$

gdzie v_1, v_2 - identyfikator zmiennej,

w_1, w_2 - identyfikator, stała lub wyrażenie.

- Instrukcja skoku

Postać ogólna instrukcji:

$$G[OTO] \langle nr \rangle$$

gdzie: nr - numer kroku realizowany po instrukcji skoku.

Numer kroku występujący w instrukcji skoku musi dotyczyć tej samej części programu (taka sama liczba całkowita numeru kroku), w której umieszczona jest instrukcja skoku.

Przykład

1.01 G 1.5 (instrukcja poprawna)

⋮

1.5 G 2.5 (instrukcja niepoprawna)

⋮

- Instrukcja pętli

Postać instrukcji umożliwiająca tworzenie pętli programowych jest następująca:

$$F[OR] \langle v \rangle = \langle w_1 \rangle : \langle w_2 \rangle : \langle w_3 \rangle \langle I_1 \rangle [\langle I_2 \rangle \langle I_3 \rangle \dots]$$

gdzie: v - wskaźnik pętli

w_1 - wartość początkowa zmiennej V,

w_2 - przyrost wartości zmiennej V,

w_3 - wartość graniczna zmiennej V ,

I_1, I_2, I_3 - instrukcje wykonywane w pętli

w_1, w_2, w_3 - mogą być stałą, zmienną, wyrażeniem.

● Instrukcja warunkowa

Instrukcje warunkowe pozwalają zmieniać sekwencje wykonywanych kroków programu. Forma instrukcji jest następująca:

$I[F] \langle R_1 \rangle [\langle R_2 \rangle , \dots] \langle I_1 \rangle [\langle I_2 \rangle \langle I_3 \rangle \dots]$

gdzie: R_1, R_2 - relacje między dwoma argumentami (symbole relacji w tab.6) lub zmienne lub $\langle x \rangle ? \langle kod \rangle :$

x - zmienna o wartości alfanumerycznej,

$kod = \begin{cases} A & - \text{weryfikacja liter} \\ D & - \text{weryfikacja cyfr} \\ P & - \text{weryfikacja jednorodności} \\ W & - \text{weryfikacja znaków} \end{cases}$

$I_1, I_2 \dots$ - instrukcje, które mają być wykonane, gdy iloczyn relacji jest TRUE.

Tab.6. Symbole relacji

Nazwa relacji	Oznaczenie
równa się	=
mniejszy	<
większy	>
mniejszy równy	<= lub =<
większy równy	>= lub =>
nie równa się	<> lub ><

spowoduje: zmianę wiersza, wydruk informacji

WPROWADZ DANE:

i wprowadzenie wartości na zmienne A i B.

● Instrukcja wyjścia

Instrukcja drukowania wyników jest postaci:

$T \langle TYPE \rangle \langle lista \rangle$

gdzie: lista - ciąg zmiennych stałych, wyrażeń arytmetycznych i znaku zmiany wiersza "!" (por. dyrektywa TYPE).

● Instrukcja komentarza

Postać komentarza jest następująca

; <dowolny ciąg znaków>

● Instrukcja wejścia

Wczytywanie wartości danych odbywa się za pomocą instrukcji o postaci:

$R[EAD] \langle lista \rangle$

gdzie: lista - ciąg zmiennych, stałych

i znaku zmiany wiersza "!", na liście nie mogą występować zmienne globalne.

Przykład

Instrukcja

3.5 READ !, "WPROWADZ DANE:", !, A, !, B

● Instrukcja zatrzymania

Instrukcja postaci:

Q[UIT]

powoduje zakończenie wykonywania programu, przechodzi się z trybu programowego do trybu natychmiastowego.

● Instrukcja wymazywania zmiennych

Postać instrukcji

K[ILL] [<v₁> , <v₂> , ...]

gdzie: v₁, v₂, ... - identyfikatory zmiennych (por. dyrektywa KILL)

● Funkcje standardowe

W programie można używać funkcji, które są definiowane w specjalnym trybie pracy systemu MUMPS.

Identyfikatory funkcji rozpoczynają się od znaku "\$".

Przykłady funkcji przedstawiono w tab.7.

Przykład

Wykreowano tablicę o identyfikatorze X:

>S x(0) = -1, X(0.01) = 2, X(131) = 100, X(99) = 132

Wartości funkcji \$HIGH będą następujące:

>T \$H(X(0)) ... najmniejszy indeks równa się 0

0.01 ... następny większy indeks

>T \$H(X(0.01))

99 ... następny indeks

>T \$H(X(99))

131 ... następny indeks

>T \$H(X(131))

- 0.01 ... wartość funkcji jeśli nie istnieje następny indeks

>T \$H(X(-0.01))

0

Tab.7. Funkcje standardowe

Lp.	Identyfikator	Opis
1	\$HIGH(X)	Argumentem funkcji jest zmienna indeksowana. Wartością funkcji jest następny indeks, do którego było odwołanie
2	\$DEFINE(X)	Argumentem funkcji jest dowolna zmienna. Wartością funkcji jest 0, jeśli zmienna nie była wykreowana w systemie lub wartość z przedz. 1-7, jeśli zmienna jest wykreowana

● Podprogramy - instrukcja DO

Instrukcja DO w programie (por. dyrektywa DO) spełnia rolę instrukcji CALL w języku FORTRAN lub GOSUB w BASIC. Postać ogólna instrukcji jest następująca:

D[0]<u₁>

gdzie: U₁ - numer kroku.

Przykład

```

:
1.1Ø T !,"OPCJE PROGRAMU"
1.2Ø T !,"W=WPROWADZENIE M=MODYFIKACJA DANYCH",!
1.3Ø R "OPCJA*", ANS I ANS="W" D 2
1.4Ø I ANS="M" D 3
1.5Ø I ANS=" " T !,"*KONIEC*",!
1.6Ø G 1.2
:

```

Powyższe instrukcje programu spowodują:

- wydrukowanie tekstu informacyjnego (krok 1.1Ø, 1.2Ø),
- wczytanie na zmienną ANS wartości danej sterującej,
- jeśli był to znak W, wykona się część 2 programu (krok 1.3Ø),
- po wykonaniu części drugiej, następną wykonywaną instrukcją będą instrukcje w krokach 1.4, 1.5, 1.6

Programowanie w języku MUMPS

Dla zilustrowania języka programowania MUMPS przedstawimy system obsługi zbioru informacji zawartych w Polskich Normach dotyczących wyrobów hutniczych. Podstawową informacją, którą wykorzystuje projektant przedstawia rys.1.

System składa się z trzech programów: NOR, GRU, KAT.

Program NOR wyprowadza dane o kątownikach do zbioru dyskowego, modyfikuje informację zmagazynowaną w zbiorze.

Program GUR odszukuje wartość parametru "grubość kątownika" przy zadanym parametrze "szerokość ramion".

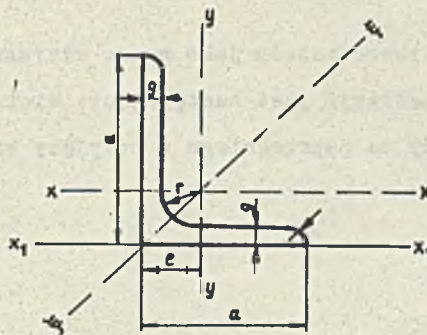
Program KAT odszukuje kątownik na podstawie zadanego momentu bezwładności I_x

Programy przechowywane są na dysku i skorzystanie z nich wymaga użycia odpowiednich dyrektyw trybu natychmiastowego (por. tryb natychmiastowy) systemu MUMPS. Są to dyrektywy ERASE, LOAD, DO.

Tekst programów zamieszczono poniżej.

Informację pierwotną (rys.1.) w programach reprezentują zmienne indeksowe globalne przechowujące wartości kolumn, i tak:

- | | |
|------------------|-------------------|
| ↑NAM - kolumna 1 | ↑IX - kolumna 9 |
| ↑A - kolumna 2 | ↑IE - kolumna 10 |
| ↑G - kolumna 3 | ↑IN - kolumna 11 |
| ↑R - kolumna 4 | ↑RIE - kolumna 12 |
| ↑R1 - kolumna 5 | ↑RIN - kolumna 13 |
| ↑P - kolumna 6 | |
| ↑M - kolumna 7 | |
| ↑E - kolumna 8 | |



PN-69/H-93401

KĄTOWNIKI RÓWNOBIAWIE

Tablica 1

Wyróżnik oznaczenia	Wymiary				Przekrój A	Masa (waga) 1 m G	Odległość od osi e	Wielkości statyczne					
	a	g	r MAX	r ₁ MAX				Moment bezwładności				Promień bezwładności	
								I _{x1}	I _{y1}	I _x	I _y	i _x	i _y
mm				cm ²	kg	cm	cm ⁴				cm		
1	2	3	4	5	6	7	8	9	10	11	12	13	
20x20x3	20	3	3,5	2	1,12	0,88	0,59	0,39	0,61	0,46	0,74	0,38	
25x25x3	25	3	3,5	2	1,42	1,11	0,71	0,80	1,26	0,33	0,94	0,48	
30x30x3	30	3	5	2,5	1,74	1,36	0,82	1,41	2,22	0,59	1,43	0,58	
30x30x4	30	4	5	2,5	2,27	1,78	0,87	1,80	2,85	0,75	1,82	0,58	
35x35x3	35	3	5	2,5	2,04	1,60	0,95	2,29	3,63	0,95	1,34	0,68	
35x35x4	35	4	5	2,5	2,67	2,09	0,99	2,95	4,68	1,23	1,32	0,68	
35x35x5	35	5	5	2,5	3,28	2,57	1,03	3,56	5,64	1,49	1,31	0,68	
40x40x3	40	3	6	3	2,35	1,84	1,06	3,45	5,45	1,44	1,52	0,78	
40x40x4	40	4	6	3	3,08	2,42	1,11	4,47	7,09	1,86	1,52	0,78	
40x40x5	40	5	6	3	3,79	2,97	1,15	5,43	8,59	2,26	1,51	0,77	
45x45x3	45	3	7	3,5	2,66	2,09	1,16	4,93	7,78	2,07	1,71	0,88	
45x45x4	45	4	7	3,5	3,49	2,74	1,22	6,43	10,48	2,68	1,71	0,88	
45x45x5	45	5	7	3,5	4,30	3,38	1,26	7,84	12,42	3,26	1,70	0,87	
50x50x3	50	3	7	3,5	2,96	2,33	1,29	6,86	10,84	2,88	1,91	0,99	
50x50x4	50	4	7	3,5	3,89	3,06	1,34	8,97	14,22	3,73	1,91	0,98	
50x50x5	50	5	7	3,5	4,80	3,77	1,39	10,97	17,38	4,55	1,90	0,97	
50x50x6	50	6	7	3,5	5,69	4,46	1,43	12,80	20,28	5,32	1,89	0,97	
60x60x5	60	5	8	4	5,82	4,57	1,63	19,37	30,71	8,03	2,30	1,48	
60x60x6	60	6	8	4	6,91	5,42	1,67	22,79	36,14	9,44	2,29	1,47	
60x60x8	60	8	8	4	9,03	7,09	1,76	29,15	46,15	12,46	2,26	1,46	
65x65x6	65	6	9	4,5	7,53	5,91	1,79	29,19	46,27	12,11	2,48	1,27	
65x65x7	65	7	9	4,5	8,70	6,83	1,83	33,43	53,00	13,87	2,47	1,26	
65x65x9	65	9	9	4,5	10,98	8,62	1,91	41,37	65,45	17,50	2,44	1,26	
75x75x5	75	5	9	4,5	7,34	5,76	1,99	38,78	61,47	16,08	2,89	1,48	
75x75x6	75	6	9	4,5	8,73	6,85	2,04	45,83	72,72	18,94	2,89	1,47	
75x75x7	75	7	9	4,5	10,10	7,93	2,08	52,61	83,49	21,73	2,88	1,47	
75x75x8	75	8	9	4,5	11,45	8,99	2,12	59,13	93,80	24,46	2,86	1,46	
75x75x9	75	9	9	4,5	12,78	10,03	2,17	65,40	103,66	27,14	2,85	1,46	
80x80x6	80	6	10	5	9,35	7,34	2,15	55,83	88,52	23,14	3,08	1,57	
80x80x8	80	8	10	5	12,27	9,63	2,24	72,25	114,61	29,89	3,06	1,56	
80x80x10	80	10	10	5	15,11	11,86	2,32	87,51	138,63	36,38	3,03	1,55	
90x90x6	90	6	11	5,5	10,57	8,70	2,34	80,32	127,29	33,35	3,47	1,78	
90x90x7	90	7	11	5,5	12,24	9,81	2,43	92,55	146,81	38,30	3,46	1,77	
90x90x8	90	8	11	5,5	13,89	10,90	2,48	104,39	165,63	43,14	3,45	1,76	
90x90x9	90	9	11	5,5	15,52	12,18	2,52	115,84	183,78	47,89	3,44	1,76	
90x90x10	90	10	11	5,5	17,13	13,45	2,56	126,92	201,28	52,56	3,43	1,75	
90x90x11	90	11	11	5,5	18,72	14,70	2,60	137,65	218,13	57,16	3,41	1,75	
100x100x8	100	8	12	6	15,51	12,18	2,72	144,85	229,82	59,88	3,85	1,96	
100x100x10	100	10	12	6	19,15	15,04	2,80	176,68	280,35	73,02	3,83	1,95	
100x100x12	100	12	12	6	22,71	17,83	2,89	206,69	327,63	85,76	3,80	1,94	
120x120x10	120	10	13	6,5	23,18	18,20	3,30	312,95	496,98	128,92	4,63	2,36	
120x120x12	120	12	13	6,5	27,54	21,62	3,38	367,67	583,71	151,63	4,60	2,35	
130x130x12	130	12	16	8	30,03	23,58	3,60	469,78	745,38	194,18	4,98	2,54	
150x150x12	150	12	16	8	34,83	27,35	4,10	736,93	1170,38	303,48	5,80	2,95	
150x150x15	150	15	16	8	43,02	33,77	4,23	898,07	1425,87	370,27	5,76	2,93	
160x160x12	160	12	16	8	37,23	29,23	4,35	902,00	1433,11	370,90	6,20	3,16	
160x160x15	160	15	16	8	46,02	36,13	4,46	1100,84	1748,77	452,91	6,16	3,14	
180x180x12	180	12	18	9	55,39	43,48	5,00	1682,45	2673,07	691,84	6,95	3,53	
180x180x15	180	15	18	9	61,91	48,60	5,08	1865,63	2962,93	768,33	6,92	3,52	
200x200x12	200	12	18	9	76,35	59,93	5,67	2850,61	4529,12	1172,09	7,70	3,92	

0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
0.11
0.12

Masę (wagę) 1 m kątowników obliczono dla wymiarów nominalnych, przyjmując gęstość stali 7,85 kg/dm³.

Rys.1. Wyciąg z normy PN-69/H-93401

Indeks (\uparrow LS) tych zmiennych interpretować należy jako numer wiersza tabelki z informacjami o produkowanych kątownikach. Sposób numeracji jest następujący: wiersz pierwszy opatrzony jest liczbą 0, a każdy następny jest większy od poprzedniego o przyjęty krok równy .01.

● Program NOR

1.01 ; CZESC PIERWSZA - WYBOR PRACY

1.05 T !, "<> PROGRAM NOR <> "

1.20 T !, "OPCJE: W=WPROWADZENIE M=MODYFIKACJA DANYCH", ! !

1.21 D 0

1.30 R "OPCJA * ", ANS I ANS="W" D 2

1.40 I ANS="M" D 3

1.50 I ANS=" " T !, "<> KONIEC-NOR<> ", ! Q

1.60 G 1.2

2.01 ; CZESC DRUGA - WPROWADZANIE DANYCH

2.10 R !, "WYROZNIK:", NAM I NAM=" " Q

2.20 S \uparrow LS= \uparrow LS+0.1, \uparrow NAM(\uparrow LS) = NAM

2.30 R !, "WYMIARY:A,G,R-MAX,R1-MAX", A,G,R,R1

2.31 R !, "PRZEKROJ:", P, "MASA:", M, !, "ODLEGLOSC OD OSI:", E

2.32 R !, "WIELKOSCI STATYCZNE", !, "MOMENT BEZWLADNOSCI", !, "IX, IY=", IX, !, "IE=", IE, !, "IN=", IN

2.33 R !, "PROMIEN BEZWLADNOSCI:", !, "RIE=", RIE, !, "RIN=", RIN!

2.35 S \uparrow A(\uparrow LS) = A, \uparrow G(\uparrow LS) = G, \uparrow R(\uparrow LS) = R, \uparrow R1(\uparrow LS) = R1

2.36 S \uparrow P(\uparrow LS) = P, \uparrow M(\uparrow LS) = M, \uparrow E(\uparrow LS) = E, \uparrow IX(\uparrow LS) = IX

2.37 S \uparrow IE(\uparrow LS) = IE, \uparrow IN(\uparrow LS) = IN, \uparrow RIE(\uparrow LS) = RIE, \uparrow RIN(\uparrow LS) = RIN

2.40 G 2.1

3.01 ; CZESC TRZECIA - MODYFIKACJA DANYCH

3.05 I \uparrow LS=0 T !, "<>NIE MA DANYCH", ! Q

3.10 S N= \uparrow LS R !, "< POPRAWA DANYCH-WYROZNIK:", NAM

3.20 I NAM=" " Q

3.22 I \uparrow NAM(N)=NAM G 3.4

3.30 S N=N-.01 I N 0 T !, "* NIE MA=>", NAM G 3.1

3.35 G 3.22

3.40 T !, \uparrow NAM(N), " ", \uparrow A(N), " ", \uparrow G(N), " ", \uparrow R(N), " ", \uparrow R1(N), " ",

\uparrow P(N), " ", \uparrow M(N), " ", \uparrow E(N), " ", \uparrow IX(N), " ", \uparrow IE(N), " ",

\uparrow IN(N), " ", \uparrow RIE(N), " ", \uparrow RIN(N)

3.50 T !, "MODYFIKACJA", !

3.51 R !, "WYROZNIK=", NAM I NAM=" " G 3.6

3.52 S \uparrow NAM(N)=NAM

3.60 R !, "A=", A I A=" " G 3.62

3.61 S \uparrow A(N)=A

3.62 R 1, "G=", G I G=" " G 3.65
3.63 S ↑G(N)=G
3.64 R 1, "R=", R I R=" " G 3.66
3.65 S ↑R(N)=R
3.66 R 1, "R1=", R1 I R=" " G 3.68
3.67 S ↑R1(N)=R1
3.68 R 1, "P=", P I P=" " G 3.70
3.69 S ↑P(N)=P
3.70 R 1, "M=", M I M=" " G 3.72
3.71 S ↑M(N)=M
⋮
3.99 G 3.1

20.01 ; CZESC CZWARTA - INICJACJA BAZY DANYCH

20.10 I \$D(↑LS)Q

20.20 S X=-.01

20.30 S ↑LS=X, X=\$H(↑NAM(X)) I. X < 0 Q

20.40 G 20.3

20.50 ; KONIEC PROGRAMU NOR

Program NOR ma za zadanie założenie bazy danych z informacjami o kątownikach równoramiennych. Część pierwsza programu (wszystkie kroki o częściach całkowitych równych 1) jest częścią sterującą i od niej powinno się zacząć wykonywanie programu.

Dyrektywa DO będzie miała postać:

DO 1

Wiersze 1.05, 1.20 spowodują wydruk:

< > PROGRAM NOR < >

OPCJE: W=WPROWADZANIE M=MODYFIKACJA DANYCH

Po wydruku (wiersz 1.30):

OPCJA *

naależy wprowadzić literę W lub M, zależnie od tego, jaką operację chce się wykonać. Wprowadzenie spacji powoduje zakończenie pracy programu, sygnalizowane wydrukiem (wiersz 1.50):

<> KONIEC - NOR <>

W wierszu 1.21 wywołana zostanie część programu ustalająca wartość początkową zmiennej globalnej ↑LS. Zmienna ta przechowuje wartość ostatniego indeksu, pod którym wpisywane były pozycje z kolumn 1-13, rys.1 do odpowiednich zmiennych globalnych.

• Program GRU

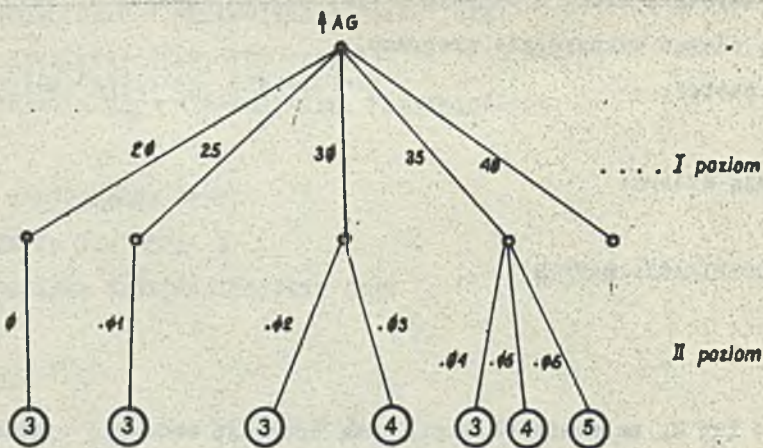
```

3.01 ;ZADANA: SZEROKOSC RAMION→POSZUKIWANA: GRUBOSC NOMINALNA
3.02 T !, "<>PROGRAM GRU<>"
3.03 I $D(↑LS) G 3.05
3.04 T !, "<>BRAK DANYCH<>KONIEC GRU<>" Q
3.05 F I=0;.01:↑LS S ↑AG(↑A(I), I)=↑G(I)
3.10 R !, "SZEROKOSC-A=", X I X=" " Q
3.20 S I=-.01
3.25 S I=$H(↑AG(X, I)) I I=>0 T !, " G=", ↑AG(X, I) G 3.25
3.30 G 3.10
3.35 ;KONIEC PROGRAMU GRU

```

W celu znalezienia grubości nominalnej kątowników o zadanych szerokościach ramion, przeprowadza się reorganizację danych pierwotnych (wiersz 3.05),

Grubości kątowników (zmienna indeksowa ↑G) przepisuje się do zmiennej globalnej ↑AG o dwóch indeksach. Jeden indeks ma wartość szerokości ramion kątownika (zmienna indeksowa ↑A), a drugi indeks wartość indeksu z poprzedniego uszeregowania danych. Powstanie w ten sposób struktura o dwóch poziomach (rys.2) gwarantująca odszukanie grubości przez odwołanie się do zmiennej ↑AG (nie ma przeszukiwania tablicy z szerokością ramion) oraz zapewniająca łączność z pozostałą częścią informacji o kątownikach przez wartość drugiego indeksu.



Rys.2. Model zmiennej AG

Program KAT

```

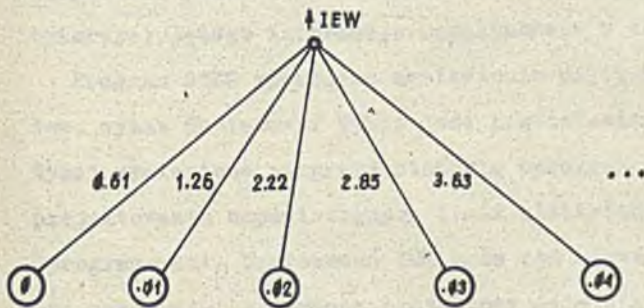
3.01 ;ZADANE: MOMENT BEZWŁADNOSCI IE→SZUKANE: KATOWNIK
3.02 T !, "<>PROGRAM KAT<>"
3.03 I $D(↑LS) G 3.05
3.04 T !, "<>BRAK DANYCH<>KONIEC KAT<>" Q
3.05 F I=0;.01: ↑LS S ↑IEW(↑IE(I)) =I
3.10 R !, "MOMENT BEZWŁADNOSCI=", X I X=" " Q
3.15 S I=-.01
3.20 S I=$H(↑IEW(I)) I I<0 Q

```



```
3.25 I I<X S IM=I G 3.20
3.30 I I=X S IW=I D 4 G 3.10
3.35 S IW=IM D 4
3.40 S IW=I D 4
3.45 G 3.10
4.01 ;DRUKOWANIE WIERSZA IW-RYS1
4.05 T I,↑NAM(IW)," ",↑A(IW)," ",↑G(IW)," ",↑R(IW)," ",
      ↑R1(IW)," ",↑P(IW)," ",↑M(IW)," ",↑E(IW)," ",↑IX(IW)," ",
      ↑IE(IW)," ",↑IN(IW)," ",↑RIE(IW)," ",↑RIN(IW)
5.01 ;KONIEC PROGRAMU KAT
```

Podobnie jak w programie GRU dokonuje się reorganizacji danych (wiersz 3.05). Kolumnę 10 przechowywaną w zmiennej indeksowej ↑IE przetwarza się za zmienną indeksową ↑IEW o wartościach indeksów równych momentowi bezwładności (rys.3).



Wartościami zmiennej indeksowej ↑IEW są indeksy z poprzedniego uszeregowania danych.

Rys.3. Model zmiennej ↑IEW

Podsumowanie

System MUMPS-11, podobnie jak system BASIC, ma wiele istotnych zalet, które powodują, że może przyczynić się do dalszego wykorzystania maszyn cyfrowych w procesie projektowania. Wprowadzenie zmiennych typu globalnego (lokowanych na dysku) w istotny sposób uprościło korzystanie z pamięci dyskowej. Zalety systemu MUMPS-11 stosowanego przy tworzeniu zbiorów informacji i wspomagającego podejmowanie decyzji przez projektanta w procesie projektowym nie mogą jednak przesłaniać niedogodności wynikających z przyjętej reprezentacji liczb w komputerach. W niektórych procesach może się okazać, że zakres liczb przedstawionych w maszynie nie gwarantuje wymaganej dokładności obliczeń projektowych.

Literatura

- [1] Bortkiewicz-Stulińska O.: Uwarunkowania wspomagania projektowania komputerem. I Krajowa Konferencja Naukowo-Techniczna, Wałbrzych 23-25.09.1980
- [2] Mocała J.: System Basic MERA-400 i WANG 2200. Porównanie. Biuletyn Informacyjny Obiektowe Systemy Komputerowe 1979 nr 4
- [3] Introduction to MUMPS-11 Language. Order No DEC-11-MALTA-C-D Digital Equipment Corporation, Maynard, Massachusetts

mgr inż. Jerzy MOCAŁA
Instytut Maszyn Matematycznych

Automatyzacja procesu projektowania. Program SPEC dla EMC MERA 400

W numerze 3/79 Biuletynu Informacyjnego Obiektowe Systemy Komputerowe został przedstawiony projekt programu KATLG, przeznaczonego do zakładania zbioru "opisów elementów" tworzącego bank danych dla potrzeb projektowania technicznego. Niniejszy artykuł zawiera opis programu SPEC wykorzystującego informacje magazynowane w zbiorze założonym za pomocą programu KATLG.

Program SPEC sporządza zestawienie użytych elementów w projektowanym obiekcie oraz sporządza tzw. wykaz montażowy. Wykaz jest przedstawieniem hierarchicznej budowy projektowanego obiektu. Wyżej wymienione programy stanowią oprogramowanie użytkowe maszyny MERA-400 przeznaczone do projektowania napowietrznych linii elektroenergetycznych. Pakiet ten opracowany w Pracowni Oprogramowania Zastosowań IMM może być używany do wspomagania innych procesów projektowych, pod warunkiem, że proces projektowy daje się zinterpretować w wymagany sposób. Rozpowszechnianiem programów zajmuje się Biuro Generalnych Dostaw Warszawskiego Centrum Naukowo Produkcyjnego Technik Komputerowych i Pomiarów.

Wprowadzenie

Podczas każdego procesu projektowania powstają opisy przedmiotu projektowanego [2]. Biorąc pod uwagę szczególność, z jaką opisy przedstawiają przedmiot projektowany w praktyce rozróżnia się następujące opisy typowe [1]:

- a) projekt wstępny - przedstawia podstawowy schemat przedmiotu, ogólny układ, podstawowe zespoły i sekcje
- b) projekt techniczny - w stosunku do opisu poprzedniego uściślone są w nim wymiary i kształty podstawowych części,
- c) projekt roboczy - przedstawia ostatecznie sprecyzowany przedmiot projektowany. Dla konstrukcji wszystkie części mają określone wymiary i kształty, tolerancje wykonawcze i jakość powierzchni.

W każdym z tych opisów przedstawia się budowę przedmiotu dokonując podziału na elementy z punktu widzenia ich złożoności. Przedmiot projektowany można interpretować jako pewną strukturę hierarchiczną o różnych poziomach złożoności elementów. Liczba poziomów złożoności zależy od przyjętego w danym procesie projektowania kryterium klasyfikacji, natomiast w każdym procesie projektowym istnieje potrzeba zliczenia elementów występujących na poziomie o najmniejszej złożoności.

Informacja ta wykorzystywana jest na każdym etapie procesu projektowania, w wyniku których powstają wyżej wymienione opisy.

I tak na etapie projektowania wstępnego informacja o liczbie elementów ostatniego poziomu złożoności (otrzymana z przetworzenia informacji zawartej w projekcie wstępnym) jest podstawą do opracowania np. harmonogramu i obliczania nakładów czasowych na dalsze etapy projektowania. Natomiast na etapie projektowania roboczego informacja ta służy do określenia planu produkcji, jeśli elementy te mają być produkowane, lub do złożenia odpowiednich zamówień, jeśli są to elementy handlowe.

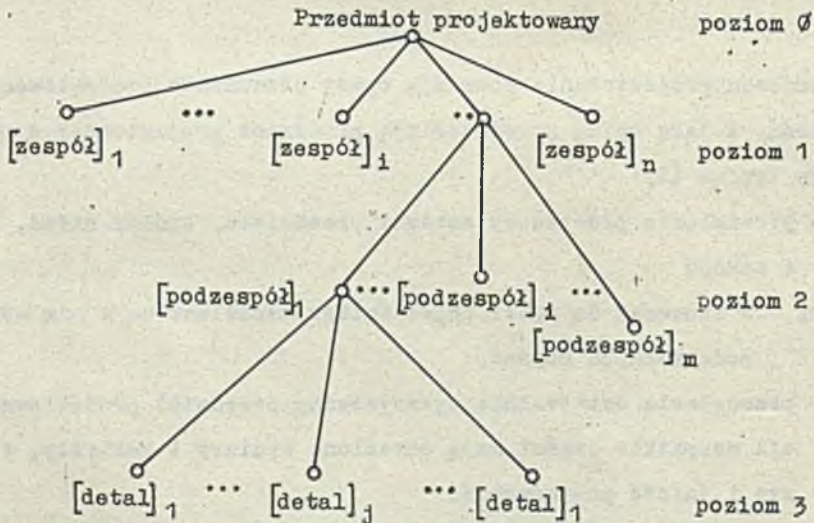
Prezentowany w artykule program SPEC dla EMC MERA-400 ma umożliwić otrzymanie przedstawionej wyżej informacji w miarę szybko i przy stosunkowo małych nakładach pracy projektanta.

Proces projektowania - wykorzystanie programu SPEC

Można założyć, że projektowany przedmiot zbudowy jest z elementów o różnej złożoności. Pod pojęciem złożoność rozumie się stopień agregacji elementów w "większe" elementy. Najczęściej rozróżnia się następujące klasy złożoności elementów:

- zespół
- podzespół
- detal

Na ogół przedmiot projektowany składa się z zespołów, zespoły z podzespołów, podzespoły z detali. Przyjmując oznaczenia konkretnych elementów należących do pewnej klasy jako: [nazwa klasy], można przedmiot projektowany przedstawić graficznie w postaci struktury hierarchicznej. Graficzną ilustracją takiej struktury jest rys.1. Na każdym poziomie rozróżnia się pewną liczbę elementów tego poziomu, co na rys.1 zostało pokazane za pomocą indeksów.



Rys. 1. Struktura przedmiotu projektowanego

Na rysunku zaznaczono przyjętą numerację poziomów złożoności. W konkretnym procesie projektowym takich poziomów złożoności może być więcej niż trzy. Poza tym może być wprowadzone dodatkowe kryterium klasyfikacji elementów, co wymagałoby użycia podwójnego indeksowania.

W procesie projektowania linii elektroenergetycznych ze względu na złożoność (jedno kryterium)

i funkcje, jakie spełnia element (drugie kryterium) rozróżnia się klasy elementów:

- zespół konstrukcji słupa
- zespół posadowienia słupa
- zespół izolacji

Elementy zwane zespołami zbudowane są z elementów prostszych - podzespołów, detali.

Projektowanie słupów po wprowadzeniu takiej klasyfikacji sprowadza się do wyboru odpowiednich elementów należących do wymienionych trzech klas zespołów. Agregacja informacji umożliwia projektantowi uzyskanie zbioru bez nadmiaru informacji, ale opis przedmiotu projektowanego na tym etapie procesu projektowania zawiera informacje o poziomach \emptyset , 1 z rys.1.

Natomiast w dokumentacji "roboczej" będącej ostatecznym opisem przedmiotu projektowanego muszą być umieszczone informacje z ostatniego poziomu złożoności, a więc wszystkie informacje konieczne do wytworzenia przedmiotu projektowanego. Aby to osiągnąć mechanicznie rozwija się strukturę elementów bardziej złożonych na elementy prostsze wg ustalonego wzoru - definicji zespołu. Praca taka staje się szczególnie uciążliwa, gdy liczba podobnych elementów jest duża.

Rozpatrując ogólnie mechanizm powstawania opisu przedmiotu projektowanego, który zawiera strukturę tego przedmiotu, można rozróżnić dwa skrajne przypadki:

- a) dekompozycję - elementy złożone rozkłada się na prostsze (postępowanie "z góry na dół")
- b) agregację - z elementów prostszych tworzy się elementy złożone (postępowanie "z dołu do góry")

W rzeczywistych procesach projektowych nie występuje tylko dekompozycja, albo tylko agregacja. Te dwa sposoby postępowania wykorzystuje się jednocześnie. Po niektórych gałęziach drzewa przedstawionego na rys.1 projektant posuwa się od dołu, a po innych do góry.

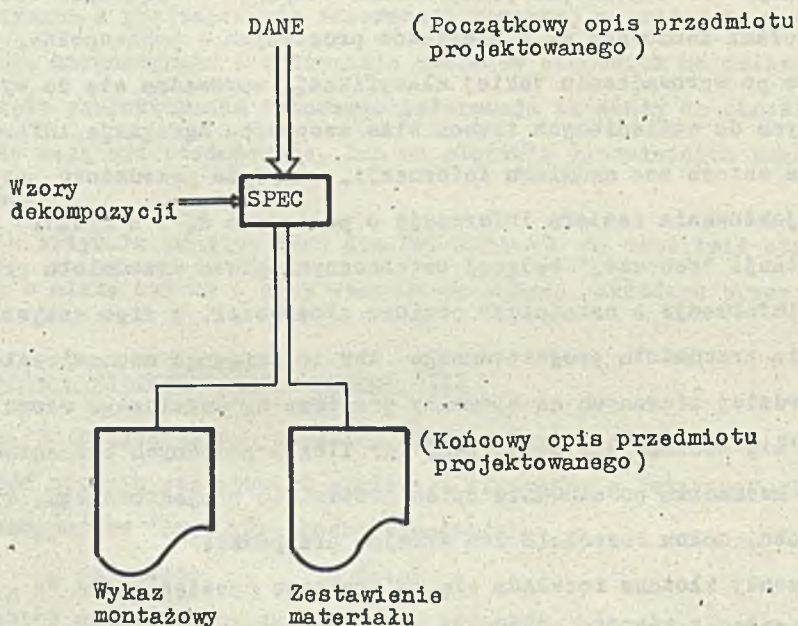
Program SPEC przeznaczony jest właściwie do rozkładania elementów złożonych na prostsze wg wzoru zapamiętanego za pomocą programu KATLG ^{x)}.

Można tu rozróżnić dwie sytuacje projektowe:

- 1) dany jest zupełny zbiór wzorów, wg których następuje dekompozycja
- 2) wzory dekompozycji definiowane są w czasie procesu projektowego

Wykorzystanie programu SPEC w sytuacji projektowej (1) obrazuje rys.2. Mając dany początkowy opis przedmiotu projektowanego np. zawierający górne poziomy złożoności \emptyset , 1, 2 (rys.1) otrzymuje się pełną strukturę przedmiotu projektowanego i wykaz montażowy oraz zestawienie elementów ostatniego poziomu złożoności (poziom 3 z rys.1) tzn. zestawienie materiału.

x) Program KATLG umożliwia magazynowanie opisów zawierających informacje o elemencie: identyfikator elementu, nazwa potoczna elementu, lista cech liczbowych, lista cech znakowych, lista elementów składowych.



Rys.2. Schemat wykorzystania programu SPEC w procesie z pełnym zbiorem wzorów dekompozycji

W przypadku stosowania programu SPEC w sytuacji projektowej (2) działanie programu SPEC musi być wspierane programem KATLG umożliwiającym uzupełnienie zbioru wzorów dekompozycji (rys.3). Na podstawie początkowego opisu przedmiotu projektowanego program SPEC dokonuje dekompozycji elementów złożonych na elementy prostsze. Oprócz tej funkcji program SPEC spełnia rolę "kontrolera" - sygnałem o braku wzoru dekompozycji jest brak opisu elementu w zbiorze wzorów. Za pomocą programu KATLG należy uzupełnić brakujące wzory i powtórnie uruchomić program SPEC. Pętlę SPEC, KATLG powtarza się tak długo, aż uzyska się końcowy opis przedmiotu projektowanego.

W dalszej części artykułu będzie ogólnie przedstawiony program SPEC, szczegółowe informacje znajdują się w dokumentacji programów SPEC [3].

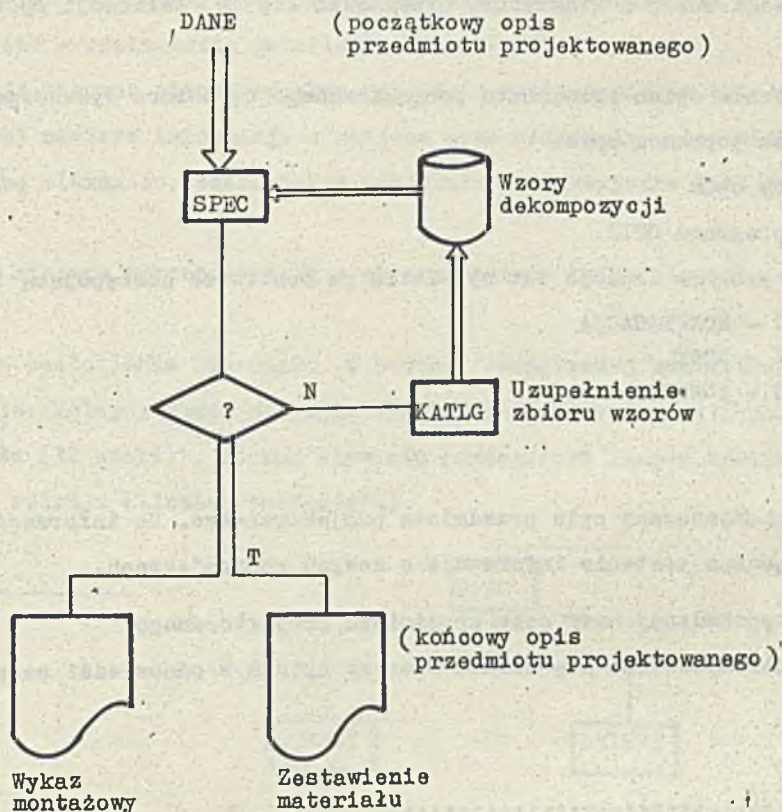
Dane dla programu SPEC

Danymi dla programu SPEC są wzorce dekompozycji (program KATLG) i początkowy opis przedmiotu projektowanego.

Początkowy opis zawiera jedynie trzy pierwsze poziomy złożoności (rys.1.) interpretowane jako:

- a) poziom 0 - nazwa przedmiotu projektowanego (projektu),
- b) poziom 1 - lista podprojektów,
- c) poziom 2 - lista elementów składowych podprojektów.

Dane te są przygotowywane za pomocą programu OPIS umożliwiający zapis informacji w zbiorze dyskowym i jej modyfikację.



Rys.3. Schemat wykorzystania programu SPEC w procesie z niepełnym zbiorem wzorów dekompozycji

Podczas realizacji programu przyjęto następujące ograniczenia dotyczące opisu przedmiotu projektowanego:

- nazwa projektu nie może zajmować więcej niż 7 wierszy po 64 znaki w wierszu
- liczba podprojektów na liście (poziom 1) zależy jedynie od wielkości zarezerwowanego obszaru na dysku przeznaczonym na zbiór, w którym przechowywane jest opisy przedmiotu projektowanego
- lista elementów składowych podprojektu nie może zawierać więcej niż 60 elementów składowych.

Tworzenie opisu początkowego przedmiotu projektowanego za pomocą programu OPIS przebiega w trybie konwersacyjnym.

Rozpoczęcie wykonywania programu OPIS sygnalizowane jest wydrukiem:

```
< * > < > PROGRAM OPIS < > < * >
ELEMENTY SKŁADOWE
**** 1 - ZAPIS
      2 - MODYFIKACJA
      3 - DRUKOWANIE ZBIORU
      4 - KONIEC
NUMER?
```


Wprowadzając odpowiedni numer z klawiatury przechodzi się do realizacji wybranej czynności, i tak:

- ① - umożliwia wprowadzenie opisu przedmiotu projektowanego do zbioru dyskowego
- ② - umożliwia dokonanie poprawek opisu
- ③ - drukuje wprowadzony opis
- ④ - kończy działanie programu OPIS.

Po podaniu numeru 1 program drukuje lub wyświetla na monitorze następującą informację:

```
< > ZAPIS < > 1 - KONTYNUACJA
                 2 - NOWY
                 3 - KONIEC
```

NUMER?

① oznacza, że będzie kontynuowany opis przedmiotu projektowanego. Do informacji zapamiętanej w zbiorze dyskowym dołączona zostanie informacja o nowych podprojektach.

② oznacza, że będzie wprowadzany nowy opis przedmiotu projektowanego

W pierwszej kolejności wprowadza się śladem wierszy tytułu w odpowiedzi na pytanie: zadane przez program:

TYTUŁ?

..... Z?

Z - numer wiersza tytułu.

Następnie program zada pytanie o nazwę podprojektu, podając nazwę pustą (spacje) kończy się wprowadzanie danych.

Elementy tworzące podprojekt wprowadza się odpowiadając na pytanie:

.....XX-----,----- ZZ ?

Jest to pytanie z o identyfikator elementu składowego (16 znaków), rodzaj (XX -liczba dwucyfrowa) i liczbę elementów.

ZZ - kolejny numer wprowadzanego elementu składowego.

Podając pusty identyfikator (spacje) elementu przerywa się wprowadzanie opisu podprojektu i przedodzi do wprowadzania opisu następnego podprojektu.

Budowa programu SPEC - funkcje segmentów programu

Program SPEC składa się z segmentu głównego i czterech podprogramów typu SUBROUTINE komunikujących się za pomocą obszarów wspólnych COMMON (rys.4).

Podział na segmenty wynika z przyjętych funkcji, jakie realizują te części programu. Funkcje segmentów są następujące:

SPEC - segment główny (MAIN); wczytuje informacje administracyjne katalogu, sprawdza, czy katalog oraz zbiór zawierający opis przedmiotu projektowanego nie jest pusty, wczytuje listę elementów podprojektu i organizuje dekompozycję elementów z listy na elementy składowe,

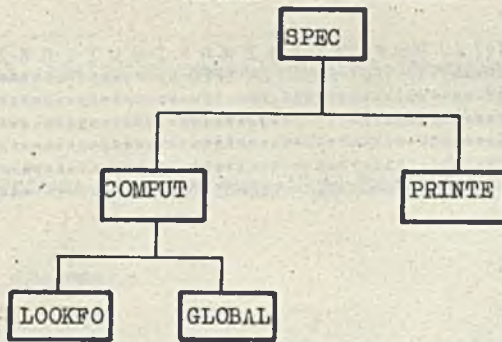
COMPUT - dekomponuje zadany identyfikatorem element na elementy składowe, aż uzyska się elementy nie mające elementów składowych (ostatni poziom złożoności), w czasie dekompozycji dru-

kuje wykaz montażowy, elementy nierozkładalne przekazywane są (GLOBAL) do zapisania na liście - zestawienie materiału,

GLOBAL - zapisuje element nierozkładalny na liście zestawienia materiału, lista w postaci maszynowej zawiera informacje o miejscu przechowywania opisu elementu w zbiorze opisów i liczbę elementów, zestawienie materiału przechowywane jest w czasowym zbiorze dyskowym

LOOKFO - ustala miejsce przechowywania w zbiorze opisów opisu elementu o zadanym identyfikatorze

PRINTE - drukuje zestawienie materiału. W postaci zewnętrznej zestawienie materiału zawiera informacje: kolejny numer pozycji, identyfikator elementu (16 znaków), nazwa potoczna elementu (32 znaki), rodzaj elementu (dwucyfrowa liczba całkowita), liczba elementów danego rodzaju (liczba rzeczywista).



Rys.4. Budowa programu SPEC

Wyniki pracy programu SPEC

W czasie rozkładania elementu na elementy składowe (por. COMPUT) tworzona jest na drukarce struktura hierarchiczna, która stanowi wykaz montażowy projektowanego obiektu. Poziom złożoności przedmiotu został skorelowany z numerem kolumny na papierze drukarki, np. elementy występujące na poziomie 10 będą drukowane począwszy od kolumny 10. Na papierze o szerokości 120 znaków, można więc uzyskać 100 poziomów hierarchii; ze względu na to, że urządzeniem wyjściowym może być monitor ekranowy w programie przyjęto możliwość przedstawienia co najwyżej 50 poziomów złożoności. W razie przekroczenia tej liczby drukowany jest komunikat na konsoli operatorskiej a elementy występujące na wyższych poziomach drukowane są począwszy od kolumny 50.

Gdy występuje element, który nie ma opisu w zbiorze opisów założonym za pomocą programu KATLG na wydruku wykazu montażowego zamiast identyfikatora elementu wydrukowane będą gwiazdki:

Na konsoli operatorskiej będzie wydrukowany komunikat:

??? XXX...X NIE MA OPISU W KATALOGU !!!

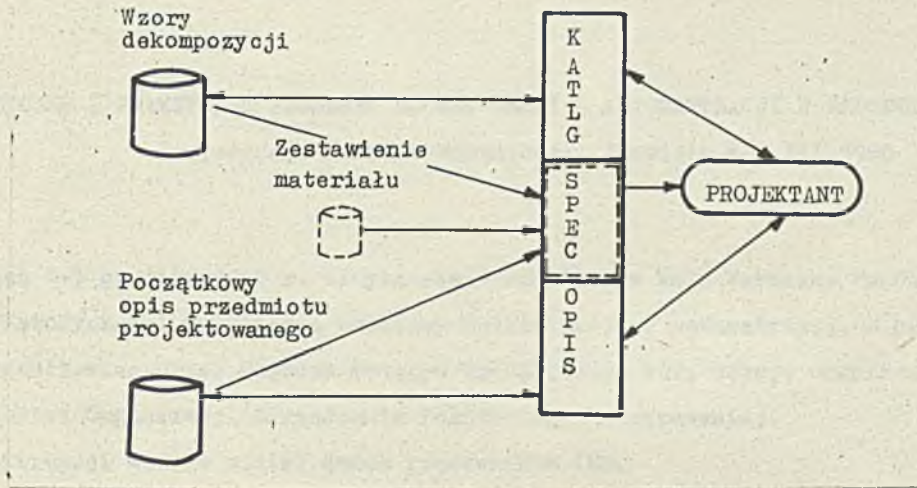
gdzie XXX...X - identyfikator elementu.

Obliczenia nie zostają przerywane, otrzyma się zestawienie materiału z wydrukowanymi gwiazdkami, zamiast identyfikatora.

W sytuacji projektowej (2) (por. rys.3) tzn. wtedy gdy wzory dekompozycji definiowane są

Organizacja pakietu dyskowego

Program SPEC jest częścią systemu przedstawionego na rys.5.



Rys.5. System wspomagający projektowanie

Przykład organizacji takiego systemu na dysku wymiennym EMC MERA-400 przedstawiono w tab.1.

Tab.1. Organizacja pakietu dyskowego

Lp.	Nazwa sekcji	Liczba sektorów	Przeznaczenie
1	AD0	2400	Przechowywanie listy cech znakowych i liczbowych
2	AD3	1200	Przechowywanie listy elementów składowych
3	AD4	1200	Procedura JCL
			Moduły ładowania programów: OPIS, SPEC, KATLG
			Zbiór 1,2 - zbiory programu KATLG
			Zbiór - początkowy opis przedmiotu projektowanego
			Zbiór - zestawienie materiału

Warto podkreślić, że specjalna procedura napisana w JCL pozwala na wywoływanie żądanych programów w sposób bardzo zbliżony do korzystania z języka problemowo zorientowanego.

Literatura

- [1] Bąbliński Cz.: Elementy nauki o projektowaniu. Warszawa WNT 1972
- [2] Mocała J.: Pewien warunek realizacji komputerowego wspomaganie projektowego. Ogólnopolska Konferencja Projektowanie III, Wrocław, 20-22.09.1978
- [3] Dokumentacja programów SPEC i OPIS. IMM, Temat R/64: Fortranowska biblioteka programów mini-komputera MERA-400.

Sprawozdania z konferencji

METODYCZNE I PRAKTYCZNE PROBLEMY MECHANIZACJI I AUTOMATYZACJI W PROJEKTOWANIU TECHNICZNYM

Konferencja naukowo-techniczna, Jadwisin 4-5.XII.1980

W dniach 4-5 grudnia 1980 r. odbyła się w Jadwisinie koło Warszawy konferencja naukowo-techniczna "Metodyczne i praktyczne problemy mechanizacji i automatyzacji w projektowaniu technicznym" zorganizowana przez Ośrodek Postępu Technicznego NOT, Sekcję Organizacji Przemysłu SIMP oraz Instytut Organizacji Zarządzania Politechniki Warszawskiej.

W konferencji wzięło udział dwóch pracowników IMM.

Na konferencji zaprezentowano 12 referatów z 14 opublikowanych w materiałach konferencyjnych i jeden nie opublikowany - wygłoszony przez dr Jagodę z CTO w Gdańsku, w którym to referacie omówiono stan komputeryzacji projektowania osiągnięty obecnie w przemyśle okrętowym, posiadany sprzęt, a mianowicie duże komputery ICL i RIAD, minikomputery PDP11, WANG oraz kalkulatory programowalne, plotery, stacje przygotowania danych. Na sprzęcie tym zaimplementowane jest oprogramowanie zarówno dla obliczeń konstrukcyjnych kształtu kadłuba, jak i dla analiz wytrzymałościowych metodą elementu skończonego, dla określania i przebiegu kabli i rurociągów, dla optymalizacji kształtu blach i dla automatycznego przygotowywania programów sterujących automatami wycinającymi blachy.

Referent przeanalizował korzyści wynikające ze stosowania nowoczesnych metod projektowania, porównał też światowe tendencje rozwojowe w dziedzinie projektowania w okrętownictwie. Zdaniem referenta obecny stan skomputeryzowania prac projektowych konstrukcyjnych i technologicznych, czyli ogólnie prac technicznego przygotowania produkcji w przemyśle okrętowym, zdecydowanie wyższy od przeciętnej krajowej, jest wymuszany przez fakt, że bez tej techniki krajowy przemysł okrętowy nie wytrzymałby konkurencji światowej. Bez systemu określania kształtu nie można by w ogóle marzyć o stawaniu do międzynarodowych przetargów, bez systemu "mes" nie można by uzyskać wpisu do rejestrów Loyda itp.

Z referatów zamieszczanych w materiałach konferencyjnych na bliższe zainteresowanie zasługują: referat doc.inż.Božo Covica z Uniwersytetu Technicznego w Mostarze (Jugosławia), pt. "Problem wartościowania pracy projektantów jako funkcja środków pracy", dr inż. Michała Trockiego z IOZ-PW pt.:"Analiza procesów projektowania technicznego z punktu widzenia ich instrumentalizacji", dr inż. Lecha Gąsiorowskiego z tego samego Instytutu pt.:"Wybrane zagadnienia doboru środków organizacyjno-technicznych dla potrzeb mechanizacji i automatyzacji procesów projektowania technicznego", mgr inż. W.Podwapińskiego z SIMP pt.: "Stan i potrzeby w zakresie zastosowań środków i urządzeń mechanizacji i automatyzacji projektowania technicznego" oraz dr Wolfganga Bartiga z Uniwersytetu Technicznego w Dreźnie pt.: "Stan aktualny oraz tendencje rozwojowe mechanizacji i automatyzacji technologicznego przygotowania produkcji w przemyśle budowy maszyn w NRD".

W dyskusji, która odbyła się na zakończenie konferencji podkreślano nieodwracalność komputerowego wspomaganie różnorodnych prac technicznego przygotowania produkcji, nie tylko prac projektowych.

Zwracano też uwagę, że równocześnie z wspomaganie komputerowym, a właściwie poprzedzając komputeryzację powinny być szeroko wprowadzone bardzo różnorodne inne środki organizacyjno-technologiczne ułatwiające prace projektowe, od szablonów i specjalistycznych suwaków, poprzez różnorodne kalkulatory, kopiarki, techniki mikrofilmowe, a dopiero po osiągnięciu tego powinno dochodzić się do wspomaganie projektowych prac przez odpowiednio programowany sprzęt komputerowy.

W odniesieniu do komputeryzacji prac projektowych zwracano w dyskusji uwagę, że oprócz niewielu silnych, a w konsekwencji przodujących w kraju ośrodków projektowych, już obecnie dobrze wyposażony w sprzęt i specjalistyczne oprogramowanie; ogromna większość małych i średnich ośrodków projektowo-konstrukcyjnych nie jest w stanie sama, o własnych siłach zorganizować przezwyciężenia się na nowoczesne metody komputerowo wspomaganego projektowania. Ośrodki takie muszą korzystać z jakiegoś poradnictwa zewnętrznego. Powinna, zdaniem licznych dyskutantów, istnieć w kraju placówka, która po zaznajomieniu się z profilem i zakresem prac danego ośrodka projektowego czy konstrukcyjnego, mogłaby zaproponować najwłaściwsze (jeden, a lepiej parę wariantowych) dla tego ośrodka sposoby i zakresy wprowadzenia komputerowo wspomaganego projektowania.

Innym ważnym, a nie rozwiązany, zadaniem dyskutantów, problemem jest ciągły brak sprawnych i operatywnych metod pełnej informacji o różnorodnych technikach i metodach usprawnienia prac projektowych o możliwościach sprzętu i oprogramowania komputerowego dostosowanego do wspomaganie tychże prac. Zdaniem dyskutantów, informacje takie nie docierają do bardzo licznych ośrodków projektowo-konstrukcyjnych.

Obie poruszane ostatnio sprawy wydają się ciekawe i dające dużo do myślenia, przy czym druga z nich może wywoływać wręcz zdziwienie. Istnieją w kraju liczne czasopisma, w których publikuje się już obecnie dużo artykułów na temat różnorodnych problemów KWP. Wydawane są też liczne katalogi oprogramowania. Istnieje wreszcie parę instytucji, które powołane były właśnie do rozpowszechniania metod nowoczesnego projektowania. Widać jednak informacja ta nie dociera czy też nie trafia do zainteresowanych.

Natomiast jednym z wniosków z konferencji było stwierdzenie, że SIMP powinien wziąć na siebie w najbliższym czasie zadanie szerokiego upowszechniania wszelkich nowoczesnych metod projektowania.

Reasumując, wydaje się, że w IMM powinno się również zauważyć problem opóźnień powszechnego stosowania metod KWP i podjąć prace wychodzące naprzeciw potrzebom małych i średnich ośrodków projektowych, które to ośrodki należałoby uznać za naturalnych potencjalnych odbiorców prac Instytutu w zakresie oprogramowania dla KWP.

NOWOCZESNE METODY PROJEKTOWANIA

I Krajowa Konferencja Naukowo-Techniczna, Wałbrzych, 23-25.IX.1980

Konferencja, zorganizowana przez SEP Wrocław, ELWRO oraz Wrocławski Oddział Polskiego Towarzystwa Cybernetycznego, odbywała się w Zamku Książ. Wzięli w niej udział pracownicy Instytutu Maszyn Matematycznych, wygłaszając dwa referaty.

Omówienie tej konferencji trzeba poprzedzić stwierdzeniem, że nazwanie jej "Pierwszą" jest mylące, gdyż jest ona pierwsza jedynie dla tego zestawu organizatorów. Trzeba natomiast pamiętać, że różnorodne konferencje na temat metodyki i metodologii projektowania odbywają się w Polsce od kilkunastu lat.

Jedną z pierwszych była zorganizowana przez PZITB w 1967 r. w Gdańsku, konferencja "ETO w budownictwie". Kontynuowane przez tych organizatorów konferencje odbywają się co parę lat w różnych miastach. Ostatnia, tzw. INFOPRO '79 odbyła się w Kudowie. Konferencje te, aczkolwiek formalnie są poświęcone komputerowo wspomaganemu projektowaniu KWP w budownictwie, to jednak z racji znacznego zaawansowania w stosowaniu technik komputerowych są ciekawe zarówno dla rozwoju KWP w innych dziedzinach, jak i dla ogólnej metodyki projektowania wspomaganego komputerowo, a wreszcie i dla wypracowywania analiz dotyczących aktualnego poziomu zapotrzebowania krajowych biur projektowych na sprzętowo-programowe narzędzia dla KWP.

Poza INFOPRO, w kraju odbywają się systematycznie następujące konferencje na temat nowoczesnych metod projektowania:

- Komitet Naukowstwa PAN zorganizował w latach 1971 i 1974 w Warszawie, a w 1978 r. we Wrocławiu (ta ostatnia pod nazwą DISIGN III) ogólnokrajowe, ale zawsze z udziałem dość licznych gości zagranicznych konferencje metodologii projektowania o zdecydowanie problemowym - metodologicznym charakterze;
- w latach 1973-77 odbyło się pięć konferencji Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej (w Międzygórzu); na konferencjach tych przedstawiciele różnych krajowych ośrodków naukowych i projektowych prezentowali zarówno problemy ogólnometodologiczne, jak i konkretne systemy wspomagające prace projektantów, nie tylko przez zastosowanie komputerów;
- nieco odmienny charakter, silniej ukierunkowany na prezentację poszczególnych rozwiązań komputerowo wspomaganego projektowania mają poznańskie "Krajowe Przeglądy Zastosowań Techniki Komputerowej w Przemśle Maszynowym", organizowane przez SIMP w cyklu dwuletnim; z tego cyklu w 1978 r. odbyła się III Krajowa Konferencja "Automatyzacja prac w projektowaniu";
- od 1977 r., co dwa lata, Instytut Podstaw Budowy Maszyn Politechniki Warszawskiej organizuje konferencje "Metody i Środki Projektowania Automatycznego"; na 1981 r. zaplanowana jest trzecia.

- również w cyklu dwuletnim, od 1976 r. organizowane są tzw. AP - konferencje naukowe na temat automatyzacji projektowania przez Politechnikę Białostocką; w konferencji tej biorą też stale udział przedstawiciele z Instytutu Politechnicznego w Kownie, z którym to Instytutem Politechnika Białostocka prowadzi stałą współpracę oraz goście z innych krajów; na konferencjach tych zawsze odbywa się pokaz systemów KWP - aktualnie realizowanych w miejscowym ośrodku obliczeniowym.
- Problemy automatyzacji i komputeryzacji projektowania występują też na wielu innych konferencjach, np. MIKRONIKA, RELKOMEX itp.

Ponadto zagadnienia nowoczesnych metod projektowania są tematem stałych, odbywających się od przeszło 10 lat w Warszawie seminariów metodologii projektowania, organizowanych przez Zakład Prakseologii, Pracownię Metodologii Projektowania IPIS-PAN - pod kierunkiem doc. W. Gasparskiego oraz seminarium ITC Politechniki Wrocławskiej pod kierunkiem prof. A. Sielickiego.

O ile pierwsze ma w zasadzie charakter ogólnokrajowy i stwarza okazję do prezentacji poszczególnym referentom ich własnych, nieraz bliskich filozofii przemysłów oraz wyników analiz o stosunkowo ogólnym znaczeniu, to drugie jest raczej imprezą ściślejszą związaną z Politechniką Wrocławską, bardziej pełni rolę seminarium szkoleniowego. Są tu prezentowane wyniki konkretnych prac teoretycznych lub aplikacyjnych.

Oprócz tych dwu ośrodków, wyraźnie metodyką projektowania zajmują się jeszcze: zespół zgrupowany wokół środowiskowego Ośrodka Informatyki Politechniki Poznańskiej, kierowanego przez doc. Z.Kierzkowskiego. W Gdańskim Centrum Techniki Okrętowej działa inż. W. Trafalski, na Politechnice Śląskiej prof. J.Dietrych i doc. W.Tarnowski, w Krakowskiej AGH prof. W.Lenkiewicz, twórca metodyki LEMACH I, II, III. W Warszawie istnieje grupa ludzi związanych z projektowaniem w budownictwie: prof. Z.Leśniak, doc. J.Szymczyk, doc. Bzymek, p. M.Wolpe i in.

Pracownicy IMM od około 6 lat aktywnie uczestniczą w tym nurcie konferencji, wnosząc swoimi referatami oryginalny sposób podejścia do zagadnień KWP, kładąc większy nacisk na problemy tworzenia oprogramowania projektującego.

Obrady konferencji w Książu rozpoczęła i zakończyła sesja ogólna, poza tym obrady toczyły się w dwu oddzielnych sekcjach, z których jedna nastawiona była bardziej na ogólne i szczegółowe problemy metodyczne, a druga na prezentowanie konkretnych rozwiązań KWP. Natomiast w materiałach konferencyjnych referaty zgrupowane są w trzech częściach.

Ogólne zagadnienia projektowania poruszane były w paru referatach o prawie filozoficznym charakterze. Jak na wielu poprzednich konferencjach, wywołały one dość żywą dyskusję, spór nawet, wokół poszczególnych terminów i sformułowań. Najwięcej i najbardziej głębokie kontrowersje wywołało wprowadzone przez prof. W.Lenkiewicza AGH pojęcie "rozwiązania idealnego" jako pewnego nierealizowalnego modelu danego rozwiązania projektowego. Koncepcję tę ostro zaatakował prof. J.Dietrych, który stwierdził, że jest to sztuczny i nic nie wnoszący formalizm, a bronił m.in. prof. Sielicki, który dowodził, że może to być wygodna koncepcja myślowa porządkująca proces twórczego koncipowania rozwiązania rzeczywistego.

Podobnie kontrowersje wywołało użycie przez jednego z referentów sformułowania: "moralne zużycie się obiektów". Prof. J.Dietrych zaprotestował przeciw używaniu tego pojęcia twierdząc, że

jest to wymysł reklamy dla napędzania koniunktury, natomiast należy tylko mówić o zużyciu technicznym. W tym też zrozumieniu powinno się dążyć właśnie do wytworzenia coraz trwalszych i bardziej niezawodnych obiektów, nie oglądając się na modę. Obrońcy istoty treści zawartych w pojęciu "zużycie moralne" wyjaśniali, że ma ono o wiele szerszy aspekt. I tak, jeżeli jakiś typ maszyn jest wprawdzie przystosowany do wymagań aktualnej technologii i mogłyby te maszyny pracować jeszcze długo i z pożytkiem, ale z istoty swojej konstrukcji wydają taki hałas, że ze względu na ludzi obsługujących te maszyny należy je zastąpić innymi, nowo- lub inaczej skonstruowanymi, ciszszymi pracującymi.

Wart również zwrócić uwagę, że ogólnie w tej grupie referatów można było zaobserwować nawrót do prostoty formy wypowiedzi i rezygnację ze zbyt daleko posuniętej formalizacji, którą odznaczały się referaty tych samych autorów na dawniejszych konferencjach. O ile uprzednio starano się często korzystać z aparatu rachunku zbiorów lub kombinatoryki itp. przy referowaniu problemów metodologicznych, o tyle na ostatniej konferencji autorzy starali się bardziej omawiać zagadnienia oraz niekiedy (może trochę za rzadko) posługiwać się schematami czy też rysunkami poglądowymi.

Tego typu formę przedstawiania problemów częściej stosowali autorzy referatów zgrupowanych w części II. Są to w zasadzie również referaty problemowe, jednak głównym problemem jest tu stosowanie technik komputerowych w projektowaniu. W tej grupie referatów na szczególniejszą uwagę zasługują wypowiedzi Kierzkowskiego, Kołodzieja, Trafalskiego oraz grupa referatów na temat grafiki komputerowej Rohatyńskiego, Sielickiego i Kruszyńskiej.

Najliczniejsze są referaty zgrupowane przez organizatorów w drugim tomie materiałów, a zatytułowanym "Część III. Metody i systemy projektowania". Wprawdzie i tu znajdują się referaty o różnym stopniu przydatności, ale większość z nich, prezentując konkretne przykłady komputerowo wspomaganego projektowania, umożliwia wyrobienie sobie własnego poglądu na rzeczywisty krajowy poziom stosowania technik komputerowych w projektowaniu. Do ciekawszych z tego punktu widzenia referatów tej grupy należy bez wątpienia zaliczyć "Projektowanie górniczych wyrobisk chodnikowych z wykorzystaniem systemów komputerowych", "Zasady eksploatacji systemu komputerowego wspomaganego projektowania APS-CNPTK1P", "Relacyjna baza danych dla projektowania statków", "Układ minikomputerowy wspomagający projektowanie rozmieszczenia elementów płaskich na powierzchniach ograniczonych", "Automatyczne projektowanie obwodów drukowanych układów cyfrowych".

Kończąc niniejsze omówienie należy podkreślić, że konferencja zgrupowała dość liczne grono ludzi stale zajmujących się problemami metodyki projektowania, równie liczną grupę ludzi dopiero wchodzących w tę problematykę oraz ludzi praktycznie zajmujących się projektowaniem, co umożliwia bezpośrednią wymianę poglądów na wiele problemów żywo obchodzących to środowisko. Dużą część referatów stanowili przedstawiciele środowisk uczelnianych, a to powodowało, że proponowane przez nich ogólne metody i koncepcje zbyt silnie związane były z dydaktyką. Było też sporo referatów wynikających z pracy biur projektowych i ośrodków obliczeniowych pracujących w tych biurach.

Wnikliwe i krytyczne prześledzenie materiałów konferencyjnych może posłużyć do wypracowania ciekawych analiz do oceny krajowego poziomu nowoczesnych metod projektowania, realnego stopnia stosowania komputerów w różnorodnych procesach projektowych, jak i prognozowania potrzeb w tym zakresie.

Założenia i warunki świadczenia usług w zakresie generacji testów
dla zakładów Zjednoczenia MERA

● Ogólne informacje o systemie

Generacja testów wchodzi w skład systemu komputerowego wspomaganego projektowania implementowanego na komputerze IRIS-80. Eksploatacja wymienionego systemu prowadzona jest w Instytucie Maszyn Matematycznych w Warszawie i obejmuje problemy związane z automatyzacją projektowania (seria dokumentacyjna systemu APS z cyklu: "Automatyzacja projektowania").

● Zakres oferty

Pracownia eksploatacji APS w ramach niniejszej oferty wykonuje następujące prace:

- a. Opracowanie testów przez SYMULACJE
- b. Generacja nośnika wyjściowego w postaci taśmy perforowanej i taśmy magnetycznej
- c. Uruchomienie opracowanego testu na urządzeniu testującym użytkownika
- d. Opracowanie postprocesora dla wyznaczonego przez użytkownika testu urządzenia testującego
Obecnie istnieją postprocesory dla testerów SAT5 i QI631P
- e. Adaptacja programów generacji testów zapewniająca bezpośrednie odczytywanie zawartości pamięci PROM z nośnika maszynowego dostarczonego przez zleceniodawcę

● Warunki techniczne generacji testów

Zgłoszone pakiety podlegają następującym ograniczeniom:

- sieć logiczna składa się z układów scalonych typu TTL małej i średniej skali integracji (SSI, MSI) z dopuszczeniem układów LSI typu pamięci RAM i PROM; aktualna lista układów scalonych symulowalnych przedstawiona jest wyżej; będzie ona rozszerzana w miarę napływających pakietów,
- liczba elementów na pakiecie nieograniczona,
- typ sieci logicznej dowolny,
- czasy propagacji sygnałów na drodze WE/WY ≤ 2 s

● Warunki realizacji

Zakłady, które zlecają do wykonania prace wymienione wyżej powinny przedstawić następujące dokumenty:

- opis funkcjonalny pakietu (przykład opisu funkcjonalnego do wglądu),
- aktualne schematy ideowe pakietu wraz z łączówką,
- wzorcowy pakiet oraz dostęp do testera w zakładzie użytkownika (dotyczy uruchomienia testu na urządzeniu testującym użytkownika)
- dokumentacja techniczna testera (opis taśmy perforowanej lub magnetycznej, opis adaptera itp.)
- dokumentacja zawartości pamięci typu PROM w wypadku nie realizowania punktu e.

Po zrealizowaniu zamówienia zakład zamawiający otrzymuje:

- ad a) Sekwencje testowe w postaci 0 i 1 na wydruku wynikowym,
- ad b) tasiemka perforowana lub magnetyczna z opracowanym testem, wydruk wynikowy opracowanego testu
- ad c) jak w pozycji b włącznie z uruchomieniem testu na ptesterze użytkownika przy wykorzystaniu pakietu wzorcowego i pomocy obsługi testera,
- ad d) w wypadku testera (dla którego nie istnieje jeszcze postprocesor) opracowanie, przy użyciu komputera R32 lub IRIS-80 nowego postprocesora

● Termin i koszty realizacji zamówienia

Od momentu dostarczenia zlecenia wraz z wymaganą dokumentacją

ad a-b - dla pakietów dużych 80-100 układów	1,5 miesiąca
dla pakietów małych 30-40 układów	1 miesiąc
ad c - dla pakietów dużych	1 tydzień
dla pakietów małych	4 dni
ad d - 3 miesiące	

Czasy realizacji zamówień podane wyżej są określone w przybliżeniu.

Koszty realizacji będą przedstawione przez Dział Ekonomiczny IMM w postaci kalkulacji wstępnej, natomiast ostateczne rozliczenie kosztu następuje wg kosztów wynikowych. Przewiduje się, że koszt całkowity dla prac z punktów a-c - nie powinien przekroczyć 150.000 zł, a dla prac z punktu d - nie powinien przekroczyć 200.000 zł.

Przewidywany termin rozpoczęcia realizacji przyjętych zamówień - 1.04.1981 r. związany ze zmianą lokalizacji komputera .

● Tryb zgłaszania zamówień

Zakłady zainteresowane przedstawioną ofertą wysyłają zamówienia na adres:

Dział Ekonomiczny i Planowania, Instytut Maszyn Matematycznych, Warszawa 08-076, ul. Krzywickiego 34, teleks 813517, telefon - centrala 21-84-41 do 49 lub 29-92-71.

Wszystkich informacji merytorycznych udziela kierownik pracowni eksploatacji APS - mgr inż. Stefan Gniazdowski, tel.23-80-01 lub 23-76-11 w.520, centrala IMM w.493.

Lista układów scalonych symulowalnych na dzień 1.08.1980 r.

1. 7400	36. 7430	71. 74LS139
2. 74H00	37. 7432	72. 74148
3. 74L00	38. 7437	73. 74150
4. 74500	39. 7438	74. 74151
5. 7401,	40. 7440	75. 74153
6. 74HC1	41. 74H40	76. 74S153
7. 7402	42. 7442	77. 74154
8. 74L02	43. 7445	78. 74155
9. 73S02	44. 7450	79. 74156
10. 7403	45. 74H50	80. 74157
11. 7404	46. 7451	81. 74S157
12. 74H04	47. 74H52	82. 74S158
13. 74L04	48. 7453	83. 74170
14. 74S04	49. 74H53	84. 74174
15. 7405	50. 7454	85. 74175
16. 7406	51. 74S64	86. 74LS175
17. 7407	52. 74565	87. 74180
18. 7408	53. 7472	88. 74181
19. 74LS08	54. 7473	89. 74182
20. 7409	55. 7474	90. 74188
21. 7410	56. 74LS74	91. 74193
22. 74H10	57. 7475	92. 74194
23. 74S10	58. 7483A	93. 74197
24. 74H11	59. 7485	94. 74200
25. 7413	60. 7486	95. 75107A
26. 7414	61. 7488A	96. 75108A
27. 74515	62. 7490	97. 75110
28. 7416	63. 7493	INTEL: 98. 3101
29. 7417	64. 7495A	99. 3212
30. 7420	65. 74107	100. 3226
31. 74S20	66. 74121	101. 3301
32. 74H21	67. 74123	102. 3601
33. 7425	68. 74125	103. 3621
34. 7426	69. 74132	ZSRR: 104. K559LP1
35. 7427	70. 74S138	105. K559LP2

WARUNKI PRENUMERATY

Prenumeratę na kraj przyjmują Oddziały RSW "Prasa-Książka-Ruch" oraz urzędy pocztowe i doręczyciele w terminie do dnia 25 listopada na rok następny.

Cena prenumeraty rocznej zł 840.

Jednostki gospodarki uspołecznionej, instytucje, organizacje i wszelkiego rodzaju zakłady pracy zamawiają prenumeratę w miejscowych Oddziałach RSW "Prasa-Książka-Ruch", w miejscowościach zaś, w których nie ma Oddziałów RSW - w urzędach pocztowych.

Czytelnicy indywidualni opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli.

Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW "Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw; ul. Towarowa 28, 00-958 Warszawa, konto PKO Nr 1153-201045.

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