

**BIULETYN TECHNICZNO-INFORMACYJNY**

P 2900/85  
KRAJOWA  
BIBLIOTEKA  
WARSZAWA

# TERMIN

PL ISSN 0239-6645  
Nr ind. 35309

**6** (276)

**1985**

Kolegium Redakcyjne:

mgr A. Chróścielewska, dr inż. J. Dyczkowski (redaktor naczelny),  
mgr J. Kutrowska (sekretarz redakcji),

Rada Programowa:

inż. J. Bartak, inż. D. Łochocki, mgr S. Majchrzak  
mgr inż. A. Musielak, inż. H. Oleksy, mgr inż. H. Piłko,  
dr inż. B. Piwowar, dr hab. inż. K. Urbaniec

**Warunki prenumeraty**

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 indywidualnie opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli. Prenumeratę roczną w cenie 1896zł należy zamawiać do 25 listopada na rok następny, półroczną do 10 czerwca na II półrocze.

Biblioteka  
Główna

9.2900/85

# **ZRZESZENIE PRODUCENTÓW ŚRODKÓW INFORMATYKI, AUTOMATYKI i APARATURY POMIAROWEJ „MERA”**



## **BIULETYN TECHNICZNO-INFORMACYJNY**

**Warszawa, czerwiec 1985**

## SPIS TREŚCI

M. Lipiński M. Owoc	Właściwości sprzężonych systemów SKOT-HADES .....	3
J. Wojdyła S. Zimnocho K. Sycz M. Owoc	Otoczenie programowe wspomagające projektowanie baz danych w systemie HADES .....	8
K. Sycz R. Nikodem A. Klar	HDSL SA - pakiet do projektowania struktur logicznych hierarchicznych baz danych w SZBD HADES .....	15
J. Wojdyła S. Zimnocho	HDSRAND - pakiet do modelowania organizacji baz danych o dostępie bezpośrednim w SZBD HADES .....	22
T. Anaszewicz M. Owoc J. Wojdyła S. Zimnocho	HDSMAP - pakiet do modelowania alokacji oraz prezentacji opisu baz danych systemu HADES .....	29
M. Warych A. Szyszka	Systemy SKOT/VS i HADES/VS - bazowe oprogramowanie narzędziowe dla m. c. JS drugiego pokolenia .....	37
Organizacja systemów informatycznych a potrzeby użytkowników.....		39

Opracowanie: Redakcja Biuletynu Techniczno-Informacyjnego "Mera", ul. Poezji 19, 04-994 Warszawa /tel. 12-90-11 wew. 17-54/. Wydawca: Ośrodek Badawczo-Wdrożeniowy MERCOMP, ul. Poezji 19, 04-994 Warszawa, Druk ZPT "Mera-Pnefal".  
Zam. 73/85, Nakład 1320 egz.

Artykuły zamieszczone w niniejszym Biuletynie MERA prezentują oprogramowanie narzędziowe dla komputerów JS EMC przeznaczone dla użytkowników, wykorzystujących w swych systemach informatyczne bazy danych. Systemy te mogą pracować w reżimie wsadowym /przy wykorzystaniu SZBD HADES/ lub w reżimie bezpośrednim /przy wykorzystaniu systemów SKOT i HADES/. Dla projektantów i programistów tych systemów przygotowano oprogramowanie pomocnicze, wspomagające procesy tworzenia i eksploatacji baz danych.

Proponujemy lekturę artykułów w następującej kolejności:

1. Właściwości sprzężonych systemów SKOT-HADES.
2. Otoczenie programowe wspomagające projektowanie baz danych w systemie HADES.
3. Pakiet HDSLSA.
4. Pakiet HDSRAND.
5. Pakiet HDSMAP.
6. Systemy HADES/VS i SKOT/VS.

Obecnie w dystrybucji są systemy SKOT, HADES, HDSMAP i systemy LABADA, SKOB i SKOZ, o których wspomina się w artykułach. Na przełomie roku 1985-86 dostępne będą systemy HDSLSA i HDSRAND, a od drugiego kwartału 1986 r. systemy SKOT/VS i HADES/VS.

Dystrybucja prowadzona jest przez: BOT ELWRO-Serwis - Dział Serwisu Oprogramowania /tel. 44-35-23/ lub Biuro Generalnych Dostaw ZE ELWRO /tel. 44-78-27/, 53-238 Wrocław, ul. Ostrowskiego 30.

Instytucją opracowującą systemy i odpowiedzialną za ich dalszy rozwój jest Instytut Komputerowych Systemów Automatyki i Pomiarów, współpracujący z Instytutem Informatyki Akademii Ekonomicznej we Wrocławiu.



mgr MIECZYŚLAW LIPIŃSKI  
IKSAiP—Wrocław  
dr inż. MIECZYŚLAW OWOC  
Akademia Ekonomiczna—Wrocław

## WŁAŚCIWOŚCI SPRZĘŻONYCH SYSTEMÓW SKOT-HADES

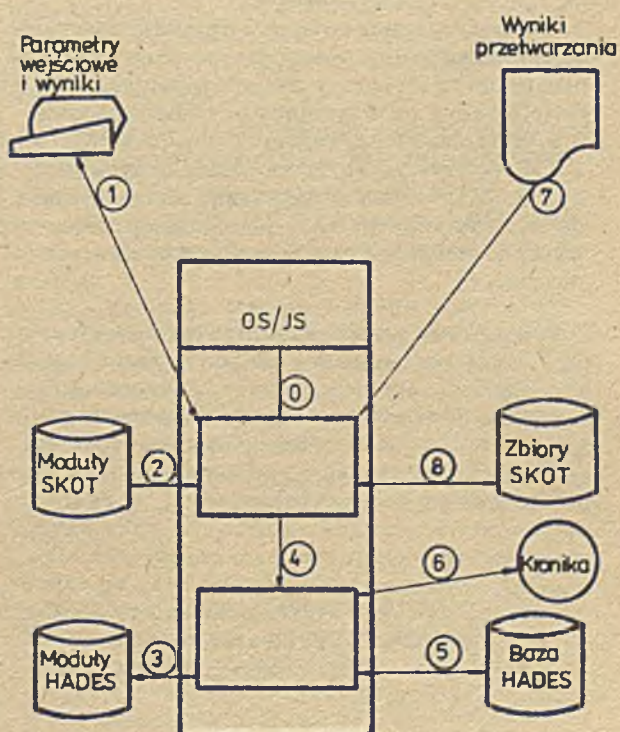
### Ogólna charakterystyka sprzężenia SKOT-HADES

Wzbogaceniem oprogramowania komputerów Serii R-32 są udostępnione w ostatnich latach Systemy SKOT i HADES. System SKOT /System Kontroli i Obsługi Terminali/ jest odpowiednikiem funkcjonalnym pakietu CICS.

System Zarządzania Hierarchiczną Bazą Danych HADES funkcjonalnie odpowiada wersji wsadowej systemu IMS/360 V. 2.

System SKOT stanowi bogatą ofertę w zakresie teleprzetwarzania, w swej standardowej postaci nie posiada jednak wielu właściwości oprogramowania zarządzającego bazą danych. Z kolei system HADES będący niezawodnym, z obszerną gamą funkcji pomocniczych, oprogramowaniem zarządzającym hierarchicznymi i gęstymi sieciami bazami danych jest przygotowany do pracy wsadowej. Istnieje jednak możliwość sprzężenia obu pakietów w jeden system o własnościach składowych zarówno jednego jak i drugiego. Artykuł opisuje taki sprzężony układ, który reprezentuje cechy nowoczesnego oprogramowania działającego w trybie bezpośrednim i wykorzystującego koncepcję technologii bazy danych. Ogólny schemat funkcjonowania sprzężonych systemów SKOT - HADES ilustruje rys. 1. Wyodrębniono na nim podstawowe zasoby wykorzystywane w sprzężeniu wraz z podstawowymi działaniami realizowanymi w omawianym układzie.

W czasie pracy układu SKOT-HADES można wyróżnić następujące rodzaje działań /na ry-



Rys. 1.

sunku zostały one zaznaczone kolejnymi numerami/:

- 0 - zainicjowanie pracy systemu SKOT przez wczytanie strumienia zadań przydzielającego podstawowe zasoby systemu,
- 1 - zainicjowanie określonych transakcji przez wywołanie z terminala odpowiedniego programu,
- 2 - wywołanie żądanych modułów i tablic systemu SKOT,
- 3 - wywołanie żądanych modułów systemu HADES /Blok Sterowania zastosowaniami, łącza DLI itp./
- 4 - realizacja odwołań systemu SKOT do systemu HADES /odwołania programowe typu CALL DLITPLI .../,
- 5 - wykonanie komend języka DLI,
- 6 - rejestrowanie zmian bazy w kronice systemu HADES,
- 7 - emisja wydawnictw na drukarkę i
- 1 - konwersacja za pośrednictwem monitora,
- 8 - współpraca ze zbiorami poza bazą danych.

Przedstawiony schemat funkcjonowania sprzężonego układu SKOT-HADES ilustruje właściwości systemu Informatycznego eksploatowanego pod nadzorem omawianego oprogramowania. Dotyczą one:

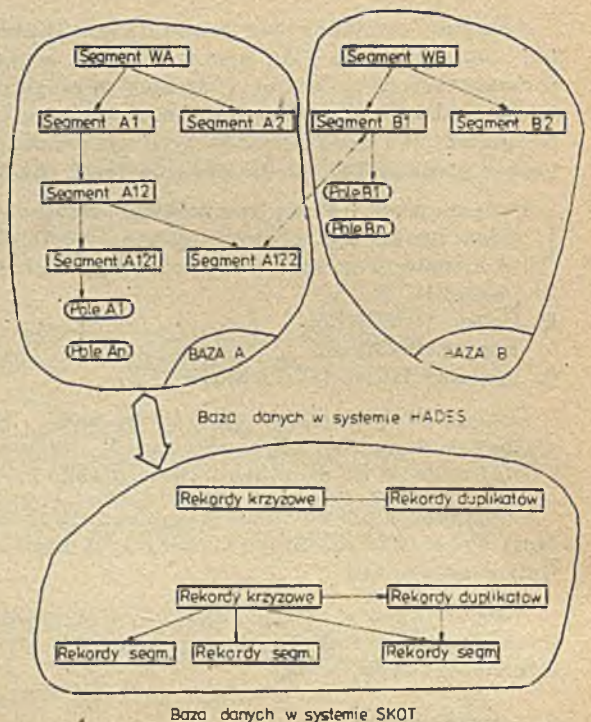
- 1/ Istoty struktur danych w bazie,
- 2/ Istoty funkcji realizowanych przez system.

Właściwości strukturalne bazy danych

Jednym z podstawowych atrybutów towarzyszących technologii bazy danych jest wyodrębnienie opisu struktury danych w centralną definicję bazy, obowiązującą przy funkcjonowaniu systemu. W sprzężonym układzie SKOT-HADES obowiązującym modelem danych jest hierarchia, w której dopuszcza się możliwość definiowania powiązań o charakterze sieciowym. Koncepcję bazy danych przedstawiono na rys. 2.

Podstawową jednostką definiowaną w strukturze bazy jest segment. Segment pełni kluczowe role w przetwarzaniu bazy, jest on bowiem obiektem oddziaływania komend manipulowania danymi. W każdej hierarchii istnieje segment nadrzędny w stosunku do pozostałych oraz grupa segmentów podporządkowanych. W ramach segmentu definiowane są pola. Na bazę danych składają się wystąpienia poszczególnych hierarchii. Właściwości struktury bazy są zaprezentowane z punktu widzenia struktury wewnątrzsegmentowej oraz zewnątrzsegmentowej. Postać graficzna opisu struktury została przedstawiona na rys. 2.

W obrębie pierwszej ze struktur występują pola. W definicji pola podaje się jego położenie /począwszy od początku segmentu/, długość oraz typ. Pola mają charakter liniowy, istnieje przy tym możliwość deklarowania obszarów



Rys. 2.

wspólnych dla kilku pól. Szczególny typ pola w segmencie, to pola kluczowe służące jako parametry lokowania segmentów, a także do konstruowania struktury zewnątrzsegmentowej. O strukturze zewnątrzsegmentowej decydują oprócz pól kluczowych także bezpośrednie łączniki adresowe. Umożliwiają one realizację określonych struktur hierarchicznych, w tym na tworzenie powiązań między segmentami różnych baz. W bazie dopuszcza się do 255 typów segmentów zdefiniowanych na 15 poziomach i umieszczonych w 10 zbiorach danych. Sygnalizowane wartości graniczne ilustrują możliwości definiowania zarówno związków wewnątrzsegmentowych jak i zewnątrzsegmentowych. Świadczy to o elastyczności strukturalnej bazy, biorąc pod uwagę poziom odwzorowań struktury bazy. Bezpośrednia definicja bazy dokonuje się za pośrednictwem tzw. Opisu Bazy Danych /DBD/, w którym określone zostają podstawowe parametry opisu hierarchii bazy. W przypadku określenia powiązań o charakterze logicznym /wykraczającym poza klasyczną strukturę hierarchiczną/ istnieje możliwość sformułowania tzw. logicznych baz danych, w których konstruować można hierarchie złożone z wielu baz elementarnych. Ostatni z poziomów odwzorowania struktury bazy to tzw. Blok Specyfikacji Programowej - PSE grupujący definicje poszczególnych baz /zarówno elementarnych jak i logicznych/ z punktu widzenia określonych zastosowań. Również na tym poziomie można dokonywać pewnych redefinicji w strukturze bazy.

Przedstawione charakterystyki wewnątrz oraz zewnątrzsegmentowe, a także kilkupoziom-

mowa koncepcja odwzorowań struktury bazy dowodzą małej niezależności danych od programów. W programie bowiem dla każdej z baz rezerwowany jest jedynie sformułowany obszar na zapamiętanie podstawowych atrybutów bazy /nazwa bazy, segmentu, poziom segmentu w strukturze, wielkość i struktura klucza itp./.

Równocześnie język, za pomocą którego definiowana jest struktura, jest językiem stosunkowo prostym i dopuszczającym określone parametry domyślne, co jest istotne dla początkujących użytkowników.

Oprócz scharakteryzowanej struktury logicznej systemu HADES w sprzężonym układzie SKOT-HADES funkcjonować może niezależnie struktura danych właściwa dla systemu SKOT. Struktura jest definiowana za pośrednictwem makroinstrukcji charakteryzujących zbiory bazy. Tworzą ją trzy typy zbiorów:

- pierwotne,
- indeksowo-krzyżowe,
- danych "duplikatów".

Pierwszy ze zbiorów "pierwotny" zawiera dane podstawowe bazy. Dane te mogą być grupowane w segmenty, blokowane lub nie o dowolnym formacie rekordu. Zbiory "indeksowo-krzyżowe" oraz danych "duplikatów" umożliwiają swobodny dostęp do zbiorów "pierwotnych". Dostęp ten uzyskuje się za pośrednictwem zawartych w zbiorach indeksach oraz tzw. "duplikatów", realizujących wielokrotne powiązanie między rekordami. Koncepcja zbiorów "pierwotnych" wraz ze zbiorami umożliwiającymi dostęp do danych w tych zbiorach umożliwia tworzenie następujących struktur danych:

- struktura prosta - zawierająca identyfikator i pola użytkowe,
- struktura posegmentowana - składająca się z segmentu podstawowego /z danymi o charakterze sterującym/ oraz z segmentów użytkowych o prostej budowie,
- struktura krzyżowa - obejmująca segmenty z identyfikatorami dla każdego z typu powiązań oraz pola użytkowe,
- struktura krzyżowo-segmentowa - łącząca właściwości obu poprzednio wymienionych,
- struktura realizująca wielokrotne powiązania za pośrednictwem zbioru "duplikatów".

Układ SKOT-HADES umożliwia także definiowanie wielu typów zbiorów o charakterze pomocniczym. W tym zbiorów związanych z obsługą zasobów systemu:

- kronika pracy systemu pozwalająca na rejestrację zmian w bazie,
- zbiory danych wewnątrzstrefowych zawierające kolejki do terminali i programów,
- zbiory danych zewnątrzstrefowych zawierające kolejki na wejściu lub wyjściu systemu,
- zbiory danych tymczasowych zawierające bloki dodatkowej pamięci zewnętrznej.

Użytkownik dysponuje dużymi możliwościami w zakresie definiowania złożonych struktur

bazy danych i korzystania ze zbiorów obsługujących zasoby sprzężonego układu.

### Właściwości funkcjonalne

#### Charakterystyka środowiska teleprzetwarzania

Podstawową właściwością systemów realizujących zadania w czasie rzeczywistym jest właściwość do takiego systemu oprogramowania teleprzetwarzania. Oprogramowanie teleprzetwarzania umożliwia eksploatację systemów informatycznych, wykorzystujących sprzęt teletransmisji, telekomunikacyjne metody dostępu itp. W sprzężonym układzie SKOT-HADES możliwe są dwa zasadnicze tryby przetwarzania:

- przetwarzanie lokalne,
- przetwarzanie zdalne.

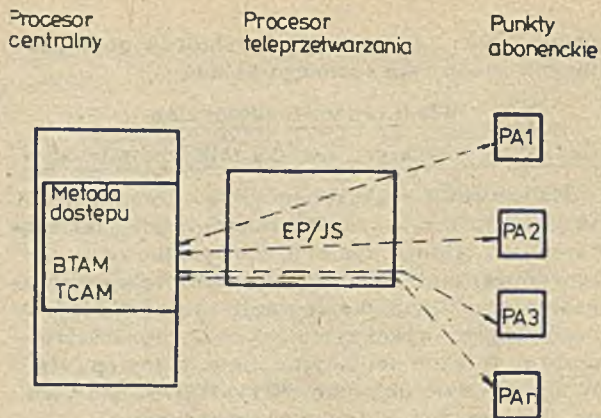
W pierwszym z przypadków sieć terminali obsługiwana przez system SKOT podłączona jest bezpośrednio do jednostki centralnej. Przy przetwarzaniu zdalnym sieć terminali współpracuje z jednostką centralną za pośrednictwem procesora teleprzetwarzania. W ramach przetwarzania lokalnego można mówić o trzech wariantach eksploatacji sprzężenia SKOT-HADES, różniących się metodą transmisji danych. W pierwszym z wariantów wykorzystywana jest metoda BTAM, która nie zezwala na pracę terminali na rzecz wielu zadań /np. SKOT i TSO/ lecz wymaga niewielkiego obszaru pamięci operacyjnej /ok. 10 KB/. Wariant eksploatacji sprzężenia SKOT-HADES, w którym używana jest metoda TCAM, umożliwiającą pracę wielu terminali na rzecz wielu zadań, lecz związane to jest ze znacznym zapotrzebowaniem pamięci operacyjnej /ok. 140 KB/. Ostatni z wariantów polega na używaniu obu z wymienionych metod /TCAM i BTAM/, łącząc właściwości poprzednich rozwiązań.

W ramach przetwarzania zdalnego konieczne jest użycie procesora teleprzetwarzania wraz z odpowiednim programem sterującym. W sprzężeniu SKOT-HADES możliwe jest użycie obu programów sterujących jednolitego systemu:

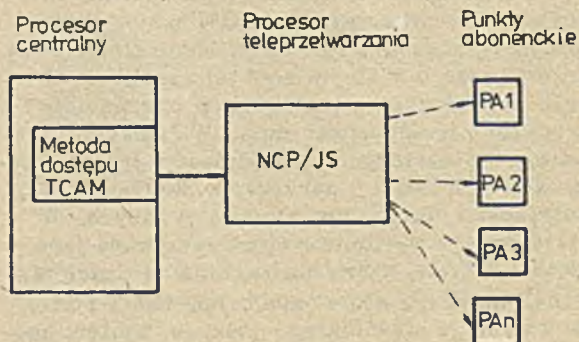
- EP - program sterujący emulujący w procesorze teleprzetwarzania funkcje multipleksera technicznego, tworzący tzw. "system buforowania znaków",
- NCP - program sterujący siecią terminalową, który przejmuje od procesora centralnego niektóre z zadań związanych z obsługą linii transmisyjnych, buforowaniem komunikatów itp.

Graficznie teleprzetwarzanie wykorzystujące oba programy sterujące ilustruje rys. 3 i 4. Użyte w ramach teleprzetwarzania programy sterujące oraz metody dostępu i transmisji danych pozwalają wyodrębnić kolejne warianty eksploatacji sprzężenia SKOT-HADES:

- program sterujący EP wraz z metodą BTAM,
- program sterujący EP wraz z metodą TCAM,



Rys. 3.



Rys. 4.

- program sterujący EP wraz z metodą BTAM i TCAM,
- program sterujący NCP wraz z metodą TCAM.

Każdy z wariantów charakteryzuje się właściwościami wynikającymi z zastosowanych składowych. Przedstawione środki programowe zarządzania siecią terminali mają możliwość obsługi wielu typów urządzeń końcowych. Są to przede wszystkim terminale: EC-7915, EC-7917, EC-8575 oraz cała grupa terminali, będących odpowiednikami funkcjonalnymi IBM /m.in. system/3, system/7, TWX rodzina serii 1000, 2000, 3000/.

#### Charakterystyka operowania danymi

Ogół komend związanych z obsługą danych w sprzężonym systemie SKOT-HADES można podzielić na dwie grupy:

- działające na danych bazy,
- obsługujące wszystkie pozostałe transmisje danych /zapisywanie i czytanie zbiorów poza bazą danych, emisja komunikatów itp./.

Język operowania danymi implementowany w systemie HADES umożliwia wykonywanie następujących czynności:

- pobieranie danych z bazy - realizowane przez różne postacie i formaty komendy GET,
- modyfikacja zawartości danych w bazie /da-

- ne w segmencie/ - realizowane za pośrednictwem komendy REPLY,
- włączanie nowych segmentów do bazy - co realizuje komenda INSERT,
- usuwanie segmentów z bazy - realizowane z wykorzystaniem komendy DELETE.

Ponadto do tego typu działań można również zaliczyć operacje tworzenia kroniki systemu. Każdej zmianie powodującej szeroką modyfikację struktury danych w bazie towarzyszy utworzenie odpowiedniego zapisu w kronice pracy systemu, zawierające wszystkie parametry realizowanej operacji /nazwa bazy, segmentu, typ komendy, parametry komendy itp./.

Żądania w zakresie obsługi zbiorów charakterystycznych dla systemu SKOT są formułowane za pośrednictwem odpowiednich makroinstrukcji. Umożliwiają one wykonanie następujących czynności podstawowych:

- wyszukiwanie danych ze zbioru /GET, SETL/.
- aktualizowanie danych w zbiorze /PUT/.
- obsługa pamięci głównej związana z zapisem nowych rekordów /RELEASE/.

Oprócz zbiorów strukturalnych bazy, obsługiwane są w ramach poszczególnych programów także wymienione poprzednio zbiory pomocnicze /dane przejściowe i strefowe/.

Obsługa dostępu do bazy może się odbywać za pośrednictwem transakcji i programów tworzonych w jednym z trzech języków bazowych: ASSEMBLER, COBOL i PL/I.

#### Charakterystyka funkcji pomocniczych

Aby ułatwić pracę projektantom i programistom systemów SKOT i HADES oraz zwiększyć niezawodność eksploatacyjną tych systemów opracowano oprogramowanie pomocnicze, wspomagające procesy tworzenia i eksploatacji baz danych, wyszukiwania informacji z baz danych, redagowania programów źródłowych itp.

Dla systemu HADES, niezależnie od funkcji pomocniczych zawartych w samym systemie /odtworzenie bazy danych, reorganizacja bazy danych, ustalanie relacji logicznych w bazie, przeglądanie bazy danych/, opracowano dodatkowe pakiety /LABADA, HDLSLA, HDSMAP, HDSRAND/ wspomagające tworzenie i eksploatację baz danych /pracujące interaktywnie pod nadzorem systemu TSO<sup>1/</sup>.

Wiele istotnych funkcji pomocniczych istnieje również w obszarze działania systemu SKOT. Przygotowane zostały następujące funkcje realizowane za pośrednictwem transakcji bądź samodzielnych pakietów:

- aktualizacja zasobów programowych,
- manualne formatowanie ekranów,
- obsługa bibliotek programów,
- obsługa zbiorów.

<sup>1/</sup> Artykuł pt. "Otoczenie programowe Systemu HADES" w niniejszym numerze Biuletynu MERA.



Realizacja funkcji - aktualizacja zasobów programowych - polega na zmianie transakcji oraz programów w trakcie eksploatacji systemu SKOT. Manualne formatowanie ekranów sprowadza się do przygotowania sformatowanego ekranu z terminala wraz z procesem kompilowania i utworzenia odpowiedniego modułu ładowalnego. Podobnie jak funkcja poprzednia może być realizowana w trakcie eksploatacji systemu bez konieczności zamykania systemu. Obsługa bibliotek programu realizowana jest z wykorzystaniem pakietu SKOB /System Konwersacyjnej Obsługi Bibliotek/. Jest to narzędzie umożliwiające wyświetlanie, modyfikowanie, kompilowanie i uruchamianie programów napisanych w jednym z bazowych języków sprzężenia SKOT-HADES: PL/I, ASSEMBLER lub COBOL.

Ostatnia z wymienionych funkcji - obsługa zbiorów - realizowana jest za pośrednictwem pakietu SKOZ /System Konwersacyjnej Obsługi Zbiorów/. Pakiet ten umożliwia wprowadzenie danych do zbiorów, które następnie mogą być wykorzystywane przez sprzężenie SKOT-HADES.

Przedstawiona charakterystyka funkcjonalna sprzężenia SKOT-HADES dowodzi wielu możliwości tego układu. Istnieje zróżnicowane środowisko teleprzetwarzania, w którym może pracować sprzężony system SKOT-HADES. Zróżnicowanie dotyczy zarówno strony sprzętowej /bogata potencjalna konfiguracja sieci terminalowej/ jak i strony programowej /metody dostępu i transmisji oraz programy sterujące/. Dość bogaty jest także mechanizm operowania danymi, szczególnie dotyczy to struktur hierarchicznych, występujących w systemie HADES /komendy DL/I wraz z prowadzeniem kroniki systemu/. Za pośrednictwem tego mechanizmu można wykonać praktycznie dowolne operacje związane z obsługą struktur bazy oraz zasobów zarządzanych przez system. Podobne wnioski można sformułować w odniesieniu do funkcji pomocniczych sprzężonego systemu SKOT-HADES.

#### Uwarunkowania sprzętowo-programowe

Systemy SKOT i HADES mogą być eksploatowane na komputerach serii RIAD, pracujących pod nadzorem systemu operacyjnego OS/JS. Minimalna wielkość pamięci operacyjnej niezbędna do pracy systemów wynosi 512 KB. Dla zainstalowania i poprawnej pracy obu systemów wystarcza standardowa konfiguracja komputerowa /wraz z urządzeniami zewnętrznymi o bezpośrednim dostępie oraz siecią terminali/. Eksploatacja sprzężenia wymaga wyposażenia instalacji w metody dostępu BTAM lub TCAM oraz w przypadku pracy na terminalach zdalnych - w procesor teleprzetwarzania wraz z programem sterującym EP lub NCP. Zapewnienie łączności programowej SKOT-HADES wymaga specjalnego generowania obu systemów. Szczegóły dotyczące tego zagadnienia

zostały podane w "Podręczniku programisty systemowego" /tom III. System SKOT-HADES/ [3]. Zasadnicze wymagania dotyczące sprzętu i oprogramowania wynikają z parametrów eksploatowanych systemów informatycznych - wielkość zastosowań /złożoność i liczba baz danych, klasy zastosowań/, wymagane czasy reakcji i niezawodność. Wymagania takie powinny być formułowane przez projektantów systemów informatycznych przy ewentualnym współdziałaniu dostawcy systemów SKOT-HADES.

#### Perspektywy układu SKOT-HADES

Prace związane z rozwojem każdego z systemów, a także ze sprzężeniem SKOT-HADES można podzielić na:

- rozwój sprzętu i oprogramowania maszyn serii RIAD,
- wzbogacanie narzędziowe obu systemów.

W pracach związanych z rozwojem sprzętu i oprogramowania maszyn serii RIAD należy uwzględnić przygotowywany model komputera R-34 dysponującego pamięcią wirtualną. Nowe możliwości związane z tą generacją maszyn zostały uwzględnione przy rozwoju sprzężenia SKOT-HADES. Opracowano nowe wersje systemów: SKOT/VS oraz HADES/VS, które wykorzystują te nowe możliwości. Należy podkreślić, że dotychczasowi użytkownicy systemów SKOT i HADES będą mieli możliwość "płynnego" przejścia na komputer R-34. Dodatkowe możliwości komputera R-34 pozwalają na podjęcie prac nad systemami przetwarzania rozproszonych baz danych, a także na realizację "relacyjnego" podejścia do danych. Oczekiwać można, że pierwszą próbą tego typu będzie materializowanie bazy relacyjnej na istniejącej bazie hierarchicznej /poprzez relacyjno-hierarchiczną metodę dostępu oraz odpowiedni język wyszukiwania informacji/.

W ramach drugiego z kierunków przygotowywane będą nowe narzędzia usprawniające proces tworzenia i eksploatacji wielodostępnego systemu bazy danych. Będą to prace związane z opracowaniem Słownika/Skorowidza Danych, a także grupę pakietów pozwalających na obserwację statystyczną i dynamiczną bazy wraz z generowaniem parametrów do strojenia bazy.

W 1985 roku przekazany zostanie system SKOT, umożliwiający realizację pytań do bazy systemu HADES oraz do zbiorów o konwencjonalnych metodach dostępu. Działanie systemu odbywa się za pośrednictwem zorientowanego na użytkownika - języka zapytań. Wyniki pracy programu emitowane są w postaci raportów lub prostych odpowiedzi. Na wyszukanych informacjach można wykonywać proste operacje logiczne i arytmetyczne.

Na podstawie przedstawionej charakterystyki układu SKOT-HADES można sformułować następujące właściwości omawianego sprzężenia:

- duże możliwości w zakresie definiowania złożonych struktur hierarchicznych z elementami powiązań sieciowych,
- łatwość manipulowania danymi,
- bogaty zestaw funkcji pomocniczych oraz narzędzi wspomagających proces tworzenia i eksploatacji systemów informatycznych, a także ułatwiających złożony proces programowania zastosowań,
- zarządzanie wielu zasobami realizujące różnorodne możliwości teleprzetwarzania,
- niezawodność systemu uzyskana dzięki wielu mechanizmom zabezpieczeń,
- łatwość obsługi,
- rozwojowość systemu,
- uniwersalność zastosowań.

Wskazane właściwości pozwalają wnioskować, że prezentowane sprzężenie SKOT-HADES jest atrakcyjnym narzędziem do obsługi wielodostępnych, telekomunikacyjnych systemów informatycznych.

#### L i t e r a t u r a :

[1] T. Anaszewicz, S. Kowalczyk, M. Owoc: Sprzężone systemy SKOT-HADES jako narzędzie

wielodostępnych, telekomunikacyjnych systemów informatycznych. Materiały z konferencji - Wspomagane komputerem tworzenie systemów informatycznych. Wrocław 1983. Prace Naukowe AE nr 228/1983.

[2] System Kontroli i Obsługi Terminali SKOT. Opis zastosowania. MERA-ELWRO, Wrocław 1979.

[3] System Kontroli i Obsługi Terminali SKOT. Podręcznik programisty systemowego. MERA-ELWRO, Wrocław 1979.

[4] System Kontroli i Obsługi Terminali SKOT. Wprowadzenie do systemu. MERA-ELWRO, Wrocław 1979.

[5] System SKOT/HADES. Oprogramowanie OS/JS. ZE ELWRO, Wrocław 1983.

[6] System Zarządzania Hierarchiczną Bazą Danych HADES. Opis zastosowań. Oprogramowanie OS/JS. ZE ELWRO, Wrocław 1983.

[7] System Zarządzania Hierarchiczną Bazą Danych HADES. Podręcznik programisty. Oprogramowanie OS/JS. ZE ELWRO, Wrocław 1983.

[8] System Zarządzania Hierarchiczną Bazą Danych HADES. Wprowadzenie. Oprogramowanie OS/JS. ZE ELWRO, Wrocław 1983.

dr JANUSZ WOJDYŁA  
mgr STANISŁAW ZIMNOCHO  
mgr inż. KAZIMIERZ SYCZ  
dr inż. MIECZYSLAW OWOC  
Akademia Ekonomiczna—Wrocław

### OTOCZENIE PROGRAMOWE WSPOMAGAJĄCE PROJEKTOWANIE BAZ DANYCH W SYSTEMIE HADES

Doświadczenia zagranicznych i krajowych użytkowników systemów zarządzania bazami danych wskazują, że jedną z najważniejszych przyczyn niepowodzeń we wdrażaniu technologii bazy danych jest brak spójnych, formalnych metod i środków programowych ułatwiających tworzenie systemu bazy danych<sup>1/</sup>. Rozumiejąc konieczność rozwiązania tego zagadnienia zespół pracowników Instytutu Informatyki Akademii Ekonomicznej we Wrocławiu od 1975 r. zajmował się problemami metodyki tworzenia baz danych. Począwszy od 1980 r. wraz z Instytutem Komputerowych Systemów Automatyki i Pomiarów /IKSAIP/ we Wrocławiu wdrożono system zarządzania hierarchicznymi bazami danych /SZBD/ HADES. Aktualna wersja HADES-SKOT umożliwia wielozadaniową realizację zadań lokalnych i zdalnych<sup>2/</sup>.

Równocześnie z rozwojem systemu rozpoczęto prace badawcze nad otoczeniem programowo-metodycznym obejmującym metodykę tworzenia i operowania hierarchicznymi bazami danych oraz narzędzia programowe wspierające ten proces. Prace te zakończyły się wdrożeniem interaktywnego laboratorium wspomaganego komputerem projektowania baz danych LABADA<sup>3/</sup>. W pierwszym okresie rozwoju laboratorium zawierało programy wspomagające pro-

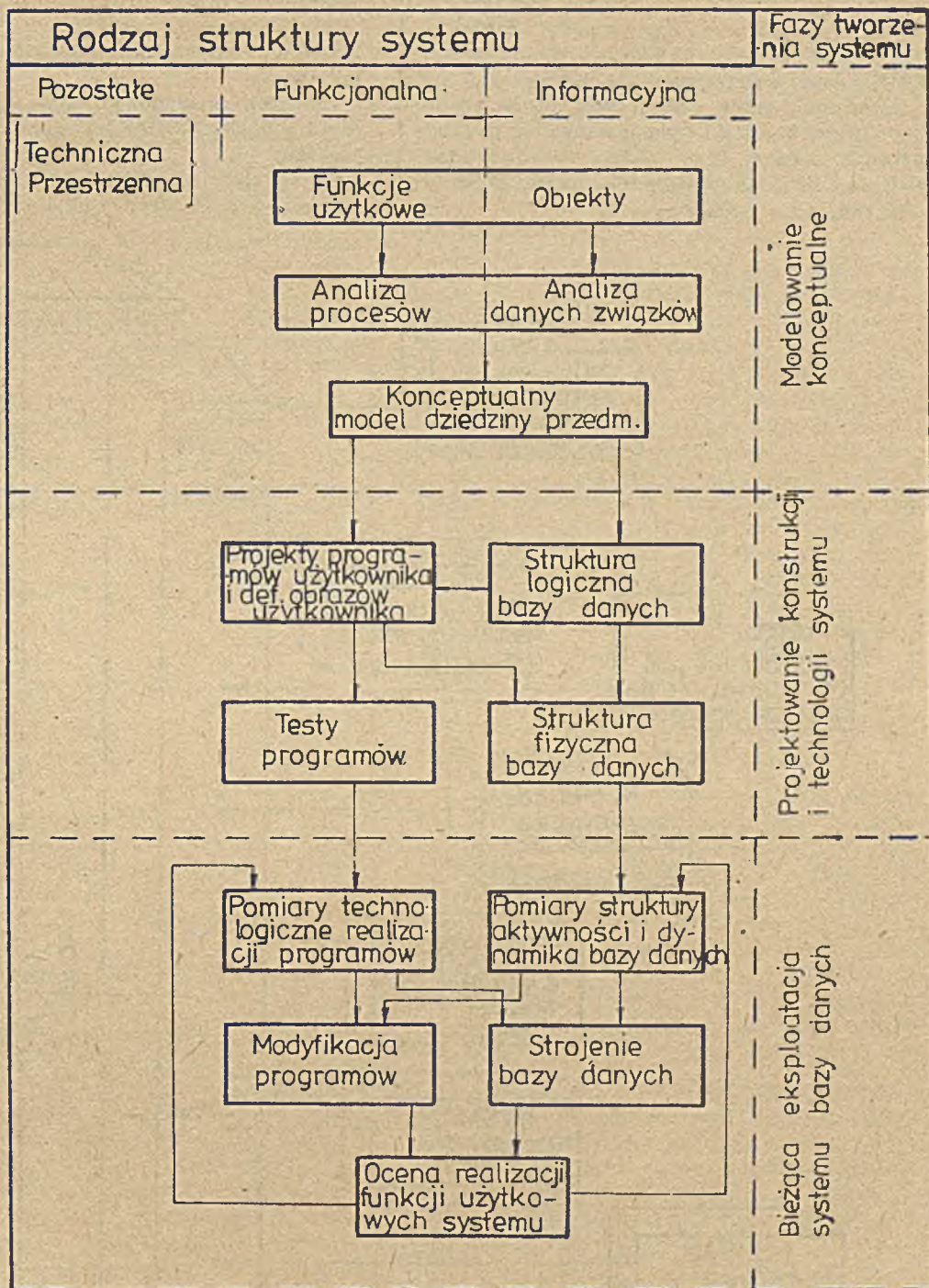
cesy analizy, modelowania, alokacji, randomizacji, itp. dla klasycznych zbiorów danych, tworzonych zgodnie ze standardami systemu operacyjnego OS/JS. W roku 1984 laboratorium zostało rozbudowane o pakiety programowe, które bezpośrednio wspomagają realizację prac nad tworzeniem informatycznych systemów zarządzania w technologii hierarchicznego SZBD HADES<sup>4/</sup>.

<sup>1/</sup> Wyniki ankiety na temat ankietowania SZBD zawarte są w artykule [2] oraz referacie C. Thanosa wygłoszonym na konferencji państw EWG [1].

<sup>2/</sup> Patrz [3] oraz pozostałe podręczniki dokumentacji systemu, a także artykuł M. Lipińskiego i M. Owoca w niniejszym numerze Biuletynu MERA.

<sup>3/</sup> Charakterystyka laboratorium LABADA jest zawarta m. in. w [4].

<sup>4/</sup> Podkreślamy tu fakt integracji wspomnianych prac w ramach laboratorium LABADA, pomimo tego, iż każdy autonomiczny pakiet może być wdrożony i eksploatowany samodzielnie.



Rys. 1. Główne elementy trójpoziomowego procesu tworzenia systemu bazy danych

Docelowy model procesu wspomaganego komputerem projektowania hierarchicznych baz danych

Przyjmując za wzorzec trójpoziomowy /trójfazowy/ model procesu tworzenia systemu informatycznego [8, 9] założono, że prace nad projektem bazy danych muszą być realizowane przez środki metodyczno-programowe integrowane pionowo /wg struktur lub procesów cząstkowych/ oraz poziomo /wg faz tworzenia systemu informatycznego/ wg podziału przedstawionego na rys. 1. Najkorzystniejszym rozwiązaniem byłoby utworzenie jednego wielodziedzi-

nowego systemu "wspomaganego komputerem tworzenia systemów informatycznych", którego zalety zostały przedstawione w pracy [10].

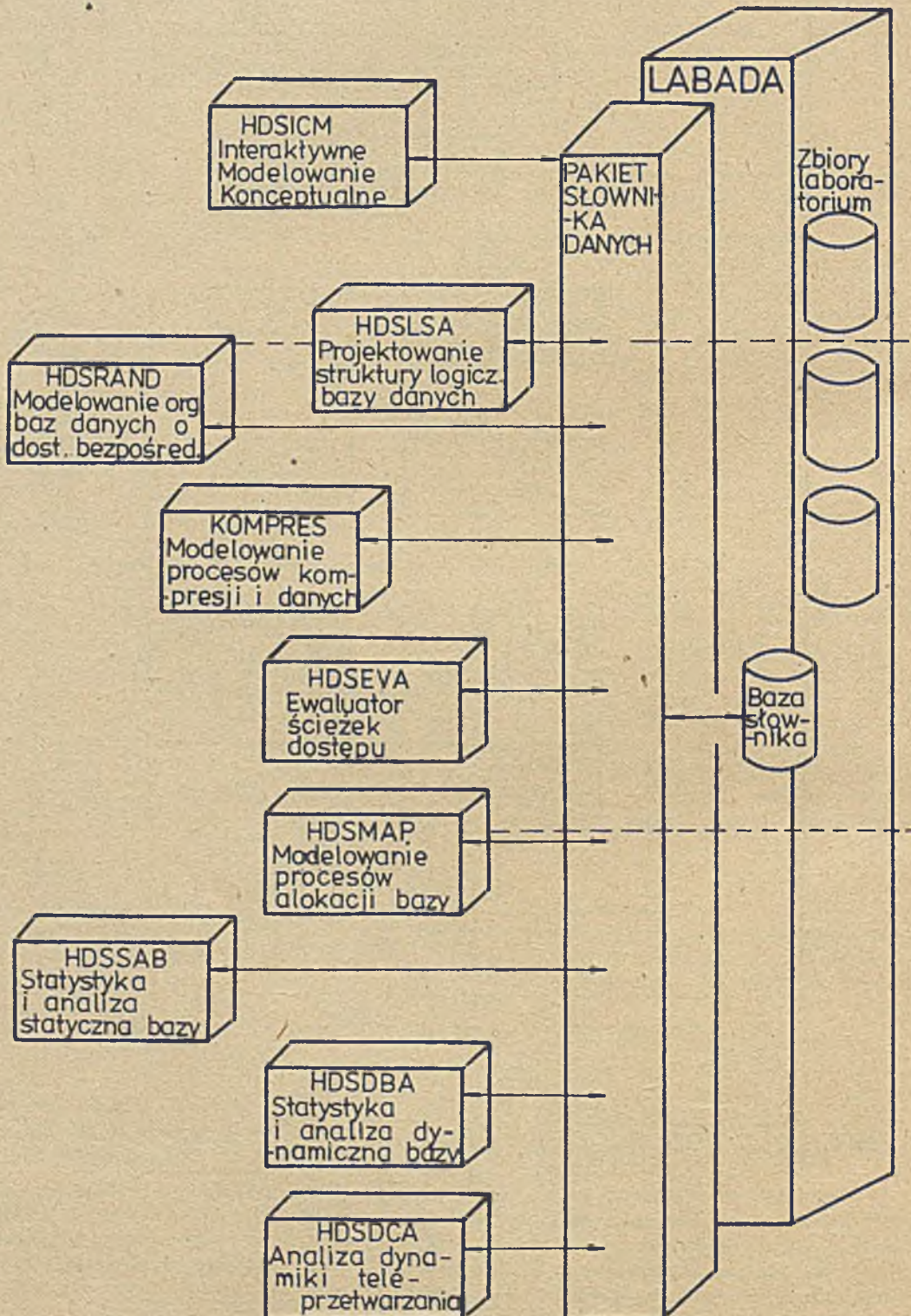
Najczęściej wyodrębniła się dwa podsystemy, z których pierwszy integruje prace nad strukturą funkcjonalną i z tej racji jest określany mianem warsztatu programisty<sup>5/</sup>. Drugi z

<sup>5/</sup> W Polsce prace takie prowadzone są w przedsiębiorstwie MERA-SYSTEM, a opisy interesujących produktów programowych, jak WPP czy SLUP zostały zawarte w artykule [7].

podsystemów integruje prace nad strukturą informacyjną, i może być określany mianem warsztatu projektanta bazy danych. Przyjmuje się, że w ramach warsztatu projektanta /WARP/ będzie funkcjonowała pewna ilość pakietów programów użytkowych, które realizując fragmenty prac wykonywanych w poszczególnych fazach, będą zintegrowane technologicznie i informacyjnie przez pakiet słownika danych, co ilustruje rys. 2.

Pakiet słownika danych /SD/ funkcjonuje we wszystkich fazach procesu tworzenia systemu informatycznego i jest:

- jedynym wiarygodnym źródłem danych projektowych,
- środkiem przechowywania i wymiany danych między uczestnikami procesu tworzenia i eksploatacji SI,
- narzędziem standaryzacji dokumentacji projektowo-programowej i eksploatacyjnej,



Rys. 2. Docelowa koncepcja wyposażenia programowego warsztatu projektanta baz danych /pogrubioną linią zaznaczono produkty aktualnie eksploatowane/

- narzędziem syntaktycznej weryfikacji obiektów słownika,
- narzędziem integrującym oba wymienione wcześniej warsztaty, tj. programisty i projektanta<sup>6/</sup>.

Prace nad słownikiem danych są zaawansowane i należy oczekiwać gotowego produktu w pierwszym lub drugim kwartale 1986 r. Laboratorium baz danych stanowi istotną pomoc w pracy projektanta i programisty. Grupuje ono różne narzędzia programowo-metodyczne, które najczęściej są prototypami przyszłych pakietów programów wspomagających, wyspecjalizowanych w rozwiązywaniu problemów tworzenia baz danych określonego typu. Funkcję takiego zaplecza prototypowni spełnia od roku 1982 interaktywne laboratorium projektowania baz danych - LABADA [4].

Najważniejszą grupę, z punktu widzenia projektanta bazy danych, stanowią programy lub pakiety programów użytkowych wspomagających realizację prac w procesie tworzenia systemu bazy danych. W kolejnych punktach zostaną scharakteryzowane główne funkcje pakietów niezbędnych w realizacji prac.

#### Narzędzia programowe konceptualnego modelowania dziedziny przedmiotowej

Zadaniem fazy konceptualnego modelowania dziedziny przedmiotowej jest adekwatny opis istotnych aspektów rzeczywistego systemu, w którym działać będzie projektowany system bazy danych. Opis ten, jako opis "rzeczywistości", ma być niezależny od aspektów technologiczno-konstrukcyjnych tworzonego systemu. Powinien precyzować np., jakie obiekty mają być reprezentowane w bazie danych, bez określania sposobu reprezentacji.

Fazę modelowania konceptualnego wyróżnia się, jako odrębny, wyraźnie określony etap, mniej więcej od 10 lat. Początkowo przyjmowano, że "istotne aspekty systemu rzeczywistego" to wyłączenie obiektów /dane/; później dołączono tu transakcje, zdarzenia i ograniczenia semantyczne dla danych i transakcji /takie jak np.: zakres dopuszczalnych wartości danych, warunki, w których można wykonać określone transakcje, itp./.. Podobny trend odnotować można w propozycjach metodycznych np. w propozycjach języków modelowania konceptualnego, jak i w stworzonych narzędziach programowych wspomagających tę fazę.

Ponieważ od momentu zgłoszenia propozycji teoretycznych do chwili wypuszczenia na ry-

<sup>6/</sup> Zainteresowanych problematyką słownika danych odsyłamy do monografii H. Lefkovitsa [11] oraz artykułów [13, 12], a także fragmentu książki [9], s. 90-98.

nek gotowego produktu programowego mijają zwykle kilka lat, dlatego też szerzej rozposzechnione narzędzia programowe fazy konceptualnej należą do "pierwszej generacji" i pomijają zwykle zarówno aspekty dynamiczne modelowania /transakcje, zdarzenia/ jak i semantyczne. Należy też dodać, że narzędzia "pierwszej generacji" łączą w sobie zwykle elementy fazy konceptualnej i logicznej /projektowanie logicznych struktur - schematów baz danych/.

Dwa najszersze stosowane produkty programowe z tej grupy to Data Designer i DBDA /Data Base Design Aid/, który jest pakietem programowym przygotowanym w firmie IBM przed rokiem 1977 [15] do współpracy z systemem IMS/VS. W fazie modelowania konceptualnego pakiet obejmuje: jednolity język opisu danych używanych przez programy użytkowe /programowi użytkowemu odpowiada coś w rodzaju "podschematu konceptualnego" czy "obrazu użytkownika"/, diagnostykę niespójności danych, w tym sprzeczności pomiędzy różnymi "podschematami", wspomaganie wykrywania synonimów i homonimów i integrację "podschematów" do wspólnego "modelu kanonicznego". Ten kanoniczny model nie jest prezentowany użytkownikowi /projektantowi/, lecz stanowi podstawę do projektowania /przez DBDA/ logicznej struktury danych, spełniającej ograniczenia systemu IMS/VS.

Podstawą do podejmowania /półautomatycznego/ decyzji dotyczących logicznej struktury danych są informacje o rodzaju przetwarzania /czytanie, modyfikacja itp./ i dane ilościowe związane z przetwarzaniem /częstotliwość realizacji danego programu, ilość dostępów do danych na jeden przebieg programu/. Oprócz tych informacji język opisu danych pozwala na wskazywanie zależności identyfikacyjnych /tzw. asocjacji/ między danymi elementarnymi.

DBDA jest pakietem wsadowym. Projektant może wpływać na jego działanie za pomocą zestawu komend. Rezultatem pracy pakietu są raporty diagnostyczne i projektowe. Istnieje łącznie programowe pozwalające na współpracę DBDA ze słownikiem/skorowidzem danych /produkcji IBM/.

Data Designer jest produktem firmy Jamesa Martina, również obejmującym projektowanie struktur logicznych [28]. Produkt nie jest związany z konkretnym systemem zarządzania bazą danych ani z modelem danych. Data Designer wykorzystuje osiągnięcia teorii relacyjnych baz danych - synteza struktur logicznych powstaje w wyniku procesu normalizacji. Opis powiązań między danymi podaje się w postaci zależności funkcjonalnych. Na prostym wyjściu graficznym /plotter/ kreślone są logiczne schematy danych w wybranym przez projektanta modelu /np. sieciowym/. Jak się wydaje, Data Designer ma nieco mniejsze możli-

wości diagnostyczne /dla fazy konceptualnej/ oraz tworzy mniej raportów wyjściowych niż opisany wyżej DBDA. Zaletami w stosunku do DBDA są: niezależność od modelu danych i względna prostota użytkowania. Data Designer integrowany jest ze słownikami danych i systemami zarządzania bazą danych - stanowi np. jeden z komponentów znanego /relacyjnego/ ADR/Datacom System.

Innym narzędziem programowym wykorzystywanym na opisywanym etapie prac nad systemem jest pakiet PSL/PSA [26]. Pozwala on na definiowanie i sprawdzanie spójności definicji zarówno danych jak i procesów na nich działających. Wykorzystuje własną wewnętrzną "bazę danych". Nie wspomaga tworzenia schematów w konkretnych /nie PSL-owskich/ modelach danych. Należy go zaliczyć do narzędzi wspomagających "pierwszej generacji".

Pewnym konkurentem dla PSL/PSA może być BSP /Business Systems Planning/ firmy IBM służący do zbierania i definiowania wymagań użytkowników /uwaga: granica pomiędzy zbieraniem i analizą wymagań a modelowaniem konceptualnym jest płynna. BSP jest jednak nie narzędziem lecz metodyką przystosowaną do słownika/skorowidza IBM [27].

W różnych ośrodkach badawczych - akademickich i przemysłowych - pracuje się nad nową generacją narzędzi wspomagających, które jak dotąd nie są szerzej dostępne na rynku informacyjnym.

Narzędzia programowe "drugiej generacji" oferują znacznie szerszy zakres udogodnień dla twórcy modelu konceptualnego. Oto zestaw cech wspólnych tego typu narzędziom:

- wyraźne oddzielenie fazy konceptualnej od innych faz tworzenia systemu,
- wzbogacony model danych do opisu schematu konceptualnego /zwykle model Entity-Relationship Chen'a, rozszerzony o relacje generalizacji czy "bycia podzbiorem"/,
- wniosek z powyższych: niezależność od docelowego SZBD,
- bezpośrednio modelowanie transakcji i zdarzeń /ułatwiający późniejsze tworzenie programów użytkowych i polepszający jakość samego schematu danych/,
- większy zakres kontroli spójności /dzięki lepszemu modelowi danych i krzyżowym testom zgodności: dane-transakcje/.

Dwa projekty badawcze, reprezentatywne dla obecnych trendów to INCOD-DTE i system GALILEO/DIALOGO, opisane np. w [16]. INCOD-DTE /Interactive Conceptual Design of Data, Translations, Events/ oparty jest na włoskiej metodologii projektowania baz danych DATAID-1 /część dotycząca bezpośrednio fazy konceptualnej opisana jest w [17]. Model Chen'a rozszerzono o generalizację między pojęciami /obiektami/ i podzbiory; o modelowanie transakcji /język pozwalający na modelowanie transakcji jest jednocześnie - podczas eksploatacji bazy - wysokiego poziomu językiem dostępu do danych/ i modelowanie zdarzeń /przy pomocy sieci Petriego/. INCOD-DTE jest systemem interakcyjnym. Model konceptualny powstaje stopniowo w wyniku wprowadzonych komend. Każde rozszerzenie spraw-

dzane jest na spójność z dotychczas utworzonym modelem. Przy wykrytej niespójności projektant otrzymuje "scenariusz" sugerujący sposoby jej usunięcia.

DIALOGO/GALILEO powstał na Uniwersytecie w Pizie. Opiera się na takim samym jak w INCOD rozszerzonym modelu danych - język opisu tego modelu jest jednocześnie wysokiego poziomu językiem ogólnego zastosowania. Nie ma odrębnego mechanizmu modelowania transakcji /lecz można je tworzyć bezpośrednio za pomocą wyrażen języka/. Ważną cechą jest możliwość opisu ograniczeń semantycznych na dane.

Najważniejszą, unikalną własnością DIALOGO jest możliwość natychmiastowego wykonania przygotowanego modelu konceptualnego. To oraz łatwość, z jaką można dostosowywać otoczenie projektowe do życzeń konkretnego użytkownika /"personalizacja łączy użytkownika"/ z nawiązką kompensującą pewną niekonwencjonalność /odejście od "technologicznego" traktowania systemu bazy danych/ systemu. Z drugiej strony owa "niekonwencjonalność" wydaje się właściwą drogą na przyszłość.

Z przedstawionych wyżej narzędzi jedynie DBDA jest w pełni zorientowany na system HADES /ze względu na "podobieństwo" systemów HADES i IMS i stał się wzorcem projektowym dla twórców pakietu HDLSLA. Z DBDA przyjęto koncepcję języka opisu danych wejściowych i ściślejszej integracji /tego fragmentu/ fazy konceptualnej z projektowaniem struktur logicznych. W przeciwieństwie do DBDA, HDLSLA jest pakietem konwersacyjnym i zawiera oryginalne rozwiązania zwiększające "inteligencję" tego narzędzia.

W Akademii Ekonomicznej we Wrocławiu trwają prace nad specyfikacjami projektowymi nowego narzędzia wspomagającego modelowanie konceptualne. Produkt programowy HDSICM będzie pierwszą realizacją wspomnianego projektu, przygotowaną do pracy w środowisku systemu HADES, zgodną z innymi narzędziami prezentowanego otoczenia programowego.

#### Narzędzia programowe wspomagające projektowanie struktury konstrukcyjno-technologicznej systemu bazy danych

Celem fazy projektowania konstrukcji i technologii przetwarzania systemu informatycznego jest zaprojektowanie struktury logicznej i fizycznej bazy danych. Wiele czynności wykonywanych przez projektanta w tej fazie może być wspomaganých za pomocą narzędzi komputerowych. Dotyczy to zwłaszcza oceny struktury danych projektowanego systemu w powiązaniu z określonymi procesami przetwarzania. Do najwcześniejszych prac w tym zakresie należy zaliczyć model FOREM opracowany i opracowany przez zespół pracowników firmy IBM pod kierunkiem M. Senko [14]. Ułatwia on ocenę funkcjonowania zbiorów danych zorganizowanych zgodnie ze standardem systemu operacyjnego OS. Wobec niewielkiej przydatności modelu FOREM w codziennej pracy projektanta, spowodowanej dużą szczegółowością modelu, powstały następane produkty programowe

do modelowania i oceny struktur danych, Wynikiem prac jest analizator organizacji zbiorów danych FDA /File Design Analyzer/ będący konkretyzacją modelu dostępu hierarchicznego S. B. Yao [19].

W drugiej połowie lat siedemdziesiątych oraz w latach osiemdziesiątych zaczęły powstawać systemy wspomagające drugiej generacji, pozwalające modelować funkcjonowanie baz danych.

Kontynuacją prac Grupy Badawczej Systemów Baz Danych z Uniwersytetu Michigan jest pakiet programów do oceny projektu bazy danych DBDE /Data Base Design Evaluator/ [23]. Pomimo że DBDE jest przeznaczony do wspomaganie prac projektowych w powiązaniu z SZBD IDS-2 firmy Honeywell, to metodologia tam wykorzystana może być rozszerzona na inne systemy sieciowe. Inną pracą zrealizowaną dla sieciowych SZBD jest pakiet EOS [24]. Obecnie jest on przystosowany do modelowania baz danych pod SZBD IDMS, lecz autorzy przewidują rozszerzenie jego możliwości na inne sieciowe SZBD - w tym RODAN. Modelowanie i ocenę hierarchicznych baz danych zarządzanych przez system IMS/VS umożliwia pakiet IBM DBPROTOTYPE II [25]. Jest to zestaw wsadowych programów umożliwiających dokonanie oceny projektu bazy danych za pomocą technik analizy statystycznej oraz symulacji.

Nieco inne podejście od wymienionych powyżej reprezentują autorzy systemu do wspomaganie decyzji projektowych w odniesieniu do projektu sieciowych baz danych [22]. Celem tego systemu jest wygenerowanie schematu wewnętrznego /fizycznego/ dla danego schematu konceptualnego. Projektant wprowadza w trybie konwersacyjnym opis schematu oraz informacje o wykorzystywanych w procesie przetwarzania ścieżkach dostępu. Opis schematu bazy danych jest generowany w języku podobnym do języka opisu danych DBTG. Oprócz problemu całościowej oceny projektu, występuje wiele zagadnień szczegółowych, których rozwiązanie wymaga stosowania złożonych technik obliczeniowych. Dwa typowe, to projekt alokacji bazy danych rozumiany jako wyznaczenie niezbędnego obszaru pamięci dyskowej dla wszystkich danych użytkownika, systemu zarządzania bazą danych i systemu operacyjnego oraz projekt bazy danych z dostępem bezpośrednim realizowanym za pomocą randomizatora. Obydwa problemy są modelowane w systemach takich jak DBDA, EOS czy DBPROTOTYPE II, lecz jedynie jako jeden z wielu elementów wpływających na funkcjonowanie bazy danych.

Koncepcja wspomaganego projektowania systemów baz danych zarządzanych przez SZBD HADES przewiduje wykorzystanie w fazie tworzenia struktury konstrukcyjno-technologicznej narzędzi programowych:

HDSRAND - do modelowania procesów randomizacji w bazach HDAM,  
HDSEVA - do modelowania ścieżek dostępu w bazach SZBD HADES,  
HDSMAP - do modelowania alokacji baz SZBD HADES,  
KOMPRES - do modelowania i realizacji kompresji danych.

Trzy spośród nich /HDSRAND, HDSMAP, KOMPRES/ znajdują się w fazie wdrażania lub eksploatacji.

#### Narzędzia programowe do oceny procesu bieżącej eksploatacji bazy danych

Ostatnia z wyodrębnionych faz tworzenia systemu /bieżąca eksploatacja/ cechuje się długotrwałością oraz koncentracją wszystkich parametrów skonstruowanego systemu /struktury informacyjnej jak i funkcjonalnej/. Konieczność tworzenia narzędzi programowych do oceny procesu bieżącej eksploatacji jest zatem bezsporna. Pomiar eksploatacyjne dokonywane przez te narzędzia programowe stają się źródłem informacji niezbędnych przy podejmowaniu decyzji modyfikujących oprogramowanie i strojenie bazy danych [18].

Obszar funkcjonalny związany z oceną procesu bieżącej eksploatacji bazy został podzielony na trzy grupy zadań:

- ocena struktur logicznych i fizycznych bazy w ujęciu statycznym,
- ocena struktur bazy i oprogramowania w ujęciu dynamicznym /dotyczy ona jedynie wersji wsadowej systemu HADES/,
- ocena struktur bazy i oprogramowania w ujęciu dynamicznym /dotyczy ona eksploatacji systemu HADES w wersji telekomunikacyjnej - w sprzężeniu z systemem SKOT/.

Narzędziem programowym korespondującym z pierwszą grupą zadań jest HDSSAB - Pakiet Statystyki i Analizy Bazy. Dostarcza ma on użytkownikowi i administratorowi bazy danych następujących informacji statystycznych:

- liczba rekordów w bazie,
- liczba segmentów danego typu,
- przeciętna liczba segmentów w rekordzie,
- struktura powiązań fizycznych segmentów w bazie /liczba i typy segmentów nad- i podrzędnych, rodzaj powiązań/,
- struktura powiązań logicznych w bazie lub między bazami /liczba i typy segmentów i powiązań logicznych/,
- przeciętna liczba segmentów podrzędnych do danego,
- liczba zbiorów bazy, zajętość zbioru,
- rozkład rekordów w blokach,
- przeciętna liczba segmentów w blokach.

Przedstawione informacje pozwalają administratorowi na korektę struktur logicznych i fizycznych bazy uwzględniającą jedynie aspekt statyczny.

Dla obsługi drugiej z grup zadań przewidziany jest pakiet HDSDBA - Analizy Dynamiki Bazy. Pozwoli on na generowanie następujących informacji eksploatacyjnych w ujęciu dynamicznym:

- charakterystyka wykonania poszczególnych komend odnoszących się do poszczególnych typów segmentów /liczba wykonań/ czas wykonania itp./,
- charakterystyka globalna odwołań do bazy /struktura komend, średni czas wykonania/,
- liczbaostępów do bazy,
- aktywność segmentów i rekordów bazy

- częstotliwość wykorzystywania określonych Bloków Specyfikacji Programowej.

Źródłem informacji dla pakietu jest prowadzona w czasie eksploatacji kronika systemu HADES. Wygenerowane przez pakiet informacje eksploatacyjne, pozwalają administratorowi na korektę struktur logicznych i fizycznych bazy uwzględniającą aspekt dynamiczny a także na podjęcie decyzji odnośnie modyfikacji oprogramowania.

Obsługę trzeciej grupy zadań ma realizować pakiet o nazwie HSDCA - Analiza Dynamiki Teleprzetwarzania, emitujący następujące informacje eksploatacyjne w ujęciu dynamicznym:

- charakterystykę programów /czas realizacji, struktura wykorzystywanych komend itp./,
- charakterystykę transakcji /czas realizacji, zaangażowane zasoby itp./,
- charakterystykę sieci /wykorzystywanie terminali, aktywność transakcji, obciążenie linii itp./.

Podstawowym źródłem danych dla pakietu jest porzerzona kronika eksploatowanego systemu. Raporty generowane przez pakiet pozwolą administratorowi na korektę sieci terminali uwzględniającą aspekty statyczne i dynamiczne bazy. Przewiduje się, że ze względu na obciążenie systemu, będzie istniała możliwość "wylączenia" pakietów z fazy gromadzenia danych eksploatacyjnych. Zakłada się duże możliwości parametryzowania wyników działania poszczególnych pakietów. Formy prezentowania raportów będą różne: od typowych wydruków tabelarycznych na drukarkę wierszową po zróżnicowane postacie wykresów na monitorze ekranowym.

#### L i t e r a t u r a :

- [1] C. Thanos: Database Design Tools, Proc. European Conference on Evaluation and Implementation of Database Systems, Sept. 25-26 1980, Brussels.
- [2] D. W. Chilson, M. E. Kudlac: Database Design: A survey of Logical and Physical Design Techniques, Data Base nr 1/1983, s. 11-19.
- [3] System Zarządzania Hierarchiczną Bazą Danych - HADES. Wprowadzenie. Oprogramowanie OS/JS. ZE ELWRO Wrocław 1982.
- [4] J. Wojdyła: Możliwości i perspektywy automatyzacji projektowania zbiorów i baz danych w ramach laboratorium LABADA. Biuletyn MERA, nr 11/1983, s. 3-8.
- [5] J. Korta, J. Wojdyła: Construction and Application of Computer - aided Data Structure Analyzer. Proc. on Sixth International Seminar on DBMS, Matrafüred-Hungary 24-29. X. 1983.
- [6] S. Zimnocho: Wspomagane komputerem projektowanie alokacji dyskowych zbiorów danych. Materiały V Konferencji Naukowo-Technicznej p.n. "Komputerowe Wspomaganie Projektowania", SIMP Poznań 1983.
- [7] K. Bulaszewski, J. Matczak, W. Mrozek: Programowe narzędzia dla produkcji oprogramowania, Biuletyn MERA, nr 7/1984, s. 21-26.
- [8] T. J. Teorey, J. P. Fry: Design of database structures, Prentice Hall 1982.
- [9] J. Oleński, W. Staniszkis: Projektowanie bazy danych. PWE Warszawa 1984.
- [10] J. Wojdyła: Modele wspomaganých kompu-

terem procesów tworzenia systemów informatycznych, Prace Naukowe AE we Wrocławiu, nr 181/1981, s. 59-68.

- [11] H. Lefkovits: Data Dictionary System. QED Monograph Series, 1977.
- [12] The British Computer Society Data Dictionary Systems Working Part Report. SIGMOD record nr 4/1977, s. 2-24.
- [13] M. E. S. Loomis, M. V. Mannino, F. W. Allen: Fundamentals of integrated Dictionary/Directory Systems. Information and Management nr 4/1981, s. 287-295.
- [14] M. E. Senko, V. Y. Lum, P. J. Owens: A File Organisation Evaluation Model /FO-REM/. Proc. of 1968 IFIP Congress, Sportan Books, Washington 1968.
- [15] N. Raven, G. V. Hubbard - Automated Logical Data Base Design: Concepts and Applications, IBM Systems Journal, vol. 16 no 3, p. 287-312.
- [16] S. Ceri /ed./, Methodology and Tools for Data Base Design, North-Holland, Amsterdam 1983.
- [17] C. Batini, M. Lenzerini: A Computer Aided Methodology for Conceptual Database Design, Information Systems, vol 7 no. 3, p. 265-280.
- [18] Charakterystyka dostępnych w firmie IBM środków wspomaganego komputerem projektowania, wdrażania i eksploatacji systemu IMS/360 v. 2 oraz IMS/VS. Raport badawczy Instytutu Informatyki AE II-3-82. Wrocław 1982.
- [19] S. B. Yao: Evaluation and Optimization of File Organizations through Analytic Modeling. Ph. D. dissertation, University of Michigan 1974.
- [20] T. J. Teorey, K. S. Das: Application of an Analytical Model to Evaluate Storage Structures. Proc. of Int. ACM/SIGMOD Conf. on Management of Data, 1976.
- [21] A. F. Cardenas: Evaluation and Selection of File Organization A Model and System. Communications of the ACM, nr 5/1975, s. 253-264.
- [22] R. Gerritsen: Steps toward the Automation of Database Design. Data Base Design Techniques I: Requirements and Logical Structures. Proc. 1978 NYU symp. on Data Base Design. Springer Verlag 1982.
- [23] T. J. Teorey, L. B. Oberlander: Network Database Evaluation Using Analytical Modeling. Proceedings of 1978 National Computer Conference, AFIPS Press, Arlington 1978.
- [24] W. Staniszkis, P. Rullo: Transaction workload analysis in the Codasyl database performance predictor EOS. Proceedings of VI International Seminar on DBMS, Matrafüred, Hungary 1983.
- [25] IBM Corp. DBPROTOTYPE II, Program Description/Operations Manual, IBM No. 5796-PJK, 1977.
- [26] M. Borchert: Język PSL - narzędziem projektanta systemów informatycznych, Informatyka 1977, nr 7.
- [27] J. G. Sakamoto, F. W. Ball, Supporting Business Systems Planning studies, w/ the DB/DC Data Dictionary, IBM Systems Journal vol. 21, no. 1, 1982.
- [28] Data Designer, 2 edition, Database Design Incorporation, June 1981.



Technologia baz danych znacznie upraszcza tworzenie systemów, skraca cykl realizacji i ułatwia rozszerzenie systemów o nowe funkcje, nie rozwiązuje jednak wszystkich trudności związanych z informatyką. Pracę projektanta baz danych można ułatwić, dostarczając mu narzędzia wspomagające. Do grupy należy pakiet HDSL SA, przystosowany do współpracy z uniwersalnym systemem zarządzania bazą danych HADES.

W dalszej części artykułu omówiono podstawowe założenia związane z pakietem HDSL SA, istotne dla projektanta struktur logicznych.

#### Projektowanie struktur logicznych

Projektowanie baz danych dzielone jest na trzy etapy: konceptualny, logiczny i fizyczny. Pierwszy z nich, związany z analizą wymagań użytkowników ma dać tzw. schemat konceptualny, wyrażający zależności między danymi w kategoriach niezależnych od stosowanego systemu zarządzania bazą danych /SZBD/. Etap logiczny ma wybrać, które z przedstawionych w schemacie konceptualnym danych będą pamiętane /fizycznie/ w bazie danych oraz przedstawić je /i zależności między nimi/ w postaci struktur stosowanych w konkretnym SZBD. Tak przedstawione dane nazwano strukturą logiczną bazy danych. Etap fizyczny to implementacja struktury logicznej.

W SZBD HADES etap logiczny obejmuje:

- wybór segmentów i ustalenie ich zawartości /pole segmentu: jego klucz, czyli tzw. pole sekwencyjne/,
- zgrupowanie segmentów w hierarchie /drzewa, struktury dopuszczalne w systemie HADES/ rekordów bazy danych,
- wskazanie relacji logicznych między segmentami,
- wskazanie indeksów pomocniczych HADES/VS pozwala na ich bezpośrednią implementację/.

Kryteria decyzji podejmowanych na etapie projektowania logicznego są następujące:

- projektowana baza danych powinna być rozszerzalna bez zmian w jej strukturze,
- powinna zawierać możliwie małą liczbę elementów redundantnych,
- powinny się w niej znaleźć tylko niezbędne relacje między danymi.

Decyzje etapu logicznego i fizycznego są od siebie uzależnione. Mimo to zaleca się, aby prace projektowe były "strukturalne", tzn. aby

przebiegały w przedstawionej wyżej kolejności /podkreślimy: dopiero po podjęciu wszystkich decyzji o strukturze logicznej bazy danych powinno rozważyć się zagadnienia jej fizycznej implementacji/. Zbyt wczesne starania o zyski w efektywności, jakie można osiągnąć przy pomocy "sprytnej" implementacji fizycznej prowadzą bowiem często do wadliwych /pod względem efektywności/ projektów.

#### Wspomaganie projektowania struktur logicznych

Projektowanie danych struktur logicznych baz danych nie jest zadaniem łatwym. Trudności z tym związane można podzielić na dwie grupy. Pierwsza grupa obejmuje problemy dotyczące analizy wymagań użytkowników. W wielu systemach wymagania różnych użytkowników są wzajemnie sprzeczne. Identyfikacje takich sprzeczności utrudnia ogrom detali, jakie jednocześnie należy brać pod uwagę.

Druga grupa trudności wiąże się z możliwością rozpatrzenia wszystkich /lub wystarczającej liczby/ alternatywnych rozwiązań sytuacji konfliktowych. Często projektant ogranicza się do jednego rozwiązania, jakie początkowo mu się nasunęło: nie zauważa innych możliwości, lub odrzuca je bez szczegółowej analizy, na którą brak mu po prostu czasu /znów nadmiar detali, jakie musiałby zauważyć/.

Pracę projektanta ułatwiają odpowiednie narzędzia, do których należy pakiet HDSL SA. Pakiet do wspomaganego projektowania logicznych struktur hierarchicznych baz danych HDSL SA /dalej będziemy używać nazwy: pakiet LSA/ ułatwia pracę projektanta baz danych w systemie HADES poprzez:

- uporządkowanie procesu projektowego /do postaci dobrze określonych podetapów, iteracyjnie powtarzanych aż do osiągnięcia zadowalającego projektu struktury logicznej/,
- wprowadzenie standardowego wzorca opisu danych, jakie mają być pamiętane w bazie /wzorca zawierającego w zasadzie wszystkie informacje niezbędne w tym etapie/,
- ochronę projektanta przed nadmiarem detali /na każdym podetapie procesu projektant może uzyskać, w postaci uporządkowanych raportów, wszystkie niezbędne w danej sytuacji informacje/,
- automatyczne podejmowanie decyzji w sytuacjach rutynowych,
- wykrywanie punktów decyzyjnych, w których

konieczna jest interwencja projektanta,

- sugerowanie potencjalnych struktur logicznych, które projektant może przyjąć lub zmodyfikować,
- wspomaganie wykrywania i rozwiązywania sprzecznych wymagań z różnych programów aplikacji,
- dokumentowanie procesu projektowania /jaki decyzje podejmowano i jaki jest ich efekt końcowy - projekt/,
- wykrywanie wszystkich potencjalnych logicznych ścieżek dostępu w tworzonej bazie /co znacznie ułatwia wprowadzanie do systemu nowych programów użytkowych/.

Pakiet LSA nie jest automatycznym generatorem optymalnych struktur logicznych baz danych, lecz narzędziem wspomagającym. Jakość zaprojektowanej bazy zależy od wiedzy i talentu projektanta, Pakiet LSA uwalnia go od części uciążliwych rutynowych zadań analitycznych, identyfikuje problemy i przedstawia alternatywne rozwiązania, ułatwia pracę projektanta i czyni ją bardziej atrakcyjną.

Użycie pakietu LSA pozwala na otrzymanie lepszych projektów końcowych i/lub znaczne skrócenie czasu poświęconego na projektowanie. Umożliwia rozważenie wielu rozwiązań alternatywnych i wybór najlepszego z nich w danej sytuacji. Pakiet LSA może być również wykorzystywany do oceny stabilności projektu /inaczej: odpowiedzi na pytania "czy określony nowy program użytkowy będzie wymuszał logiczną reorganizację bazy?"/. Może być też cennym narzędziem kształcenia i doskonalenia projektantów baz danych.

#### Pojęcia podstawowe do modelowania danych

Dane przechowywane w bazie danych są odzwierciedleniem wybranych cech obiektów i pojęć ze świata rzeczywistego, w którym powstaje system informatyczny z bazą danych. Jednakże nie wszyscy użytkownicy /nie wszystkie programy użytkowe/ są zgodni co do granic tych obiektów i pojęć, co do zbioru przysługujących im cech. Projektant bazy danych musi te granice /te zbiory cech/ określić jednoznacznie i precyzyjnie, uwzględniając wymagania użytkowników. W tym celu musi posługiwać się terminem lepiej zdefiniowanym niż "obiekt". Podstawowym dlań pojęciem jest "dana elementarna".

**D a n a e l e m e n t a r n a** - to podstawowa jednostka informacji, stwierdzająca fakt o obiekcie czy pojęciu. Przykładowo - interesuje nas obiekt, jakim jest osoba. Nazwisko = Kowalski, wiek = 40, stanowisko = dyrektor, to trzy fakty o tej osobie, a więc trzy dane elementarne. Dana elementarna jest jednostką atomową, logicznie niepodzielna. Dana elementarna składa się z nazwy i wartości.

● **N a z w a** - to symboliczne odwołanie do wszystkich wystąpień tej danej elementarnej. Nazwa określa rodzaj interesującego nas faktu. W naszym przykładzie nazwami danych elementarnych są: nazwisko, wiek i stanowisko.

● **W a r t o ś ć** - musi być powiązana z nazwą, aby coś znaczyć. Wartość oznacza konkretny fakt. Kowalski, 40 i dyrektor to trzy wartości, trzy fakty o pewnej osobie, trzy wystąpienia danej elementarnej.

● **W y s t ą p i e n i e** - to wartość danej elementarnej dla konkretnego obiektu.

Na etapie projektowania projektanta interesują nie konkretne wartości, lecz nazwy danych elementarnych. Tej samej danej elementarnej /temu samemu rodzajowi faktu/ można nadać różne nazwy. Zadaniem projektanta jest m.in. rozwiązanie konfliktu nazw /synonimy/ - jednoznaczne przyporządkowanie nazw rodzajom faktów /rodzajom danych elementarnych/.

Rozróżniamy dwie klasy danych elementarnych: klucze i atrybuty.

**Klucz** - to dana elementarna o unikalnych wystąpieniach, której wartości pozwalają na jednoznaczną identyfikację innych /zwykle kilku/ danych elementarnych z nim powiązanych. W świecie rzeczywistym klucz pozwala na rozróżnienie obiektów czy pojęć tego samego rodzaju. **Atrybut** - to dana elementarna, dostarczająca dodatkowych informacji o obiekcie identyfikowanym przez klucz. Ma zwykle wartości nieunikalne.

W systemie HADES klucz wraz ze wszystkimi atrybutami, które identyfikuje tworzy segment.

Dla każdego z programów operujących na bazie danych można podać listę danych elementarnych, z których będzie on korzystał. Niektóre pozycje tej listy identyfikują inne dane elementarne. Zależność identyfikacji między dwiema danymi elementarnymi nazwano asocjacją.

**Asocjacja** - to jednokierunkowy związek identyfikacji między dwiema danymi elementarnymi. Dana "od" identyfikuje /tzn. precyzuje znaczenie/ danej "do". Mówimy, że dana "od" jest w tej asocjacji identyfikatorem, dana "do" - daną identyfikowaną. Każde z wystąpień identyfikatora precyzuje znaczenie jednej, kilku lub zera wystąpień danej elementarnej "do", zależnie od typu asocjacji.

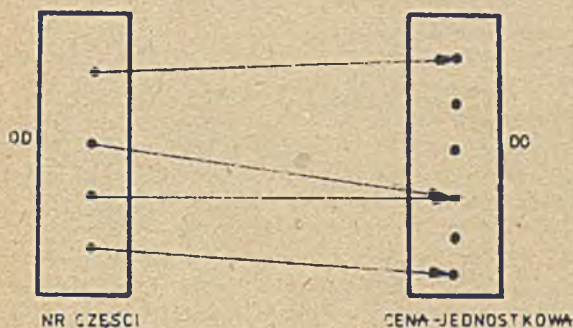
Przykładowo: dwie dane elementarne - nazwisko i wiek tworzą asocjację prostą. Nazwisko jest identyfikatorem dla wieku. Dla każdego wystąpienia nazwisko /np. Kowalski z podanego przykładu/ pojawi się dokładnie jedno wystąpienie wieku tegoż osobnika /tu: 40/. Wystąpienie wieku "40" jest w ogólnym przypadku nieunikalne. Jest ono precyzowane przez asocjację, na podstawie której wiemy, że "40" odnosi się do Kowalskiego.

### Typy asocjacji

Wyróżniamy trzy typy asocjacji:

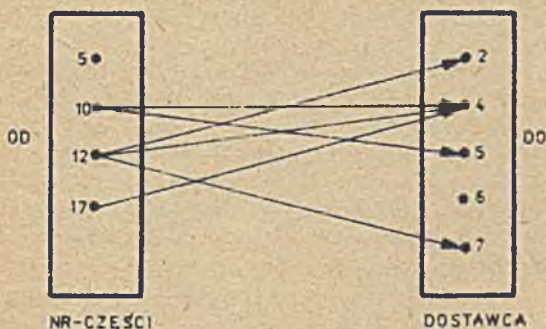
- prosta - typ "1",
- złożona - typ "M",
- warunkowa - typ "C".

Asocjacja prosta /1/ to taka, w której każde wystąpienie danej elementarnej "od" identyfikuje dokładnie jedno wystąpienie danej elementarnej "do". Przykładowo asocjacja NR-CZĘŚCI; CENA-JEDNOSTKOWA jest typu 1, gdyż każda z części ma tylko jedną cenę /zakładamy, że jest tak bez względu na dostawcę itp./.



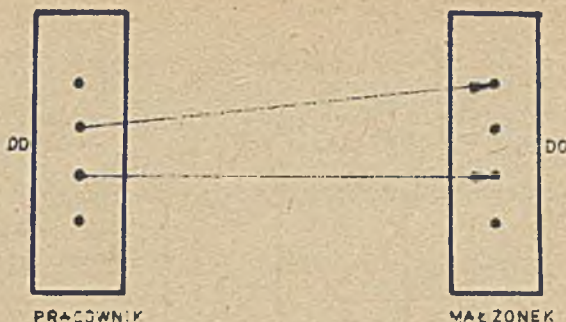
Rys. 1. Asocjacja prosta

Asocjacja złożona /M/, to taka, w której każde wystąpienie danej elementarnej "od" identyfikować może dowolną ilość /w tym zero/ wystąpień elementarnej "do". Przykładowo asocjacja NR-CZĘŚCI; DOSTAWCA jest zwykle asocjacją typu M, gdyż w ogólnym przypadku określona część może być dostarczana przez kilku dostawców.



Rys. 2. Asocjacja złożona

Asocjacja warunkowa /C/ przypomina asocjację typu 1, z tą różnicą, że wystąpienie "do" może nie istnieć. Każde z wystąpień danej "od" identyfikuje albo dokładnie jedno albo nie identyfikuje żadnego wystąpienia danej elementarnej "do". Typowym przykładem asocjacji warunkowej jest zależność PRACOWNIK; MAŁŻONKA. Pracownik może być kawalerem lub żonatym; w tym drugim przypadku ma tylko jedną żonę.



Rys. 3. Asocjacja warunkowa

Danymi wejściowymi dla pakietu LSA są, dla każdego z programów użytkowych, listy danych elementarnych, z których korzysta program, z zaznaczonymi związkami pomiędzy tymi danymi. Związki te podawane są w postaci asocjacji. Dla każdej tak powiązanej pary danych elementarnych podawany jest typ asocjacji. Informacje te - nazwy danych i typy asocjacji - podawane są na wejściu pakietu LSA za pośrednictwem "formularza specyfikacji zapotrzebowania". Formularz ten jest dokumentem standaryzującym postać informacji zbieranych na etapie analizy i modelowania konceptualnego. Dla każdego użytkownika /zastosowania/ przygotowany jest zwykle jeden formularz. Obok wspomnianych danych zapisywane są na nim informacje związane z charakterystyką przetwarzania, a więc jak często wykonywany jest dany program, ile przeciętnie danych elementarnych wykorzystuje się w przetwarzaniu /względem danej identyfikującej/, czy określona dana jest czytana, modyfikowana itp. Informacje te są wykorzystywane przez pakiet LSA przy podejmowaniu niektórych decyzji strukturalnych. Formularze dla poszczególnych zastosowań mogą być przygotowywane niezależnie od siebie. Upraszcza to analizę i modelowanie konceptualne. Ewentualne niespójności i sprzeczności, jakie mogą pojawić się pomiędzy różnymi zastosowaniami zostaną wykryte, a w pewnych wypadkach usunięte przez pakiet LSA.

### Tworzenie odwzorowań

Pojęcie asocjacji jest szczególnie ważne dla pakietu LSA. Na podstawie podanych na wejściu asocjacji pakiet ten może automatycznie dokonać klasyfikacji danych elementarnych na klucze i atrybuty, pogrupować dane elementarne w segmenty oraz określić hierarchiczne powiązania między segmentami. W pewnych wypadkach pakiet korzysta z faktu, że określona para danych elementarnych może być powiązana dwoma asocjacjami - role "od" i "do" pełnione są przez obie dane na przemian.

Odwzorowanie - to para asocjacji łączących dwie dane elementarne. Jedną z asocjacji nazywamy umownie asocjacją "naprzód", drugą - asocjacją "wstecz". Jest przy tym sprawą

umowną, który z kierunków asocjacji przyjmujemy jako "naprzód".

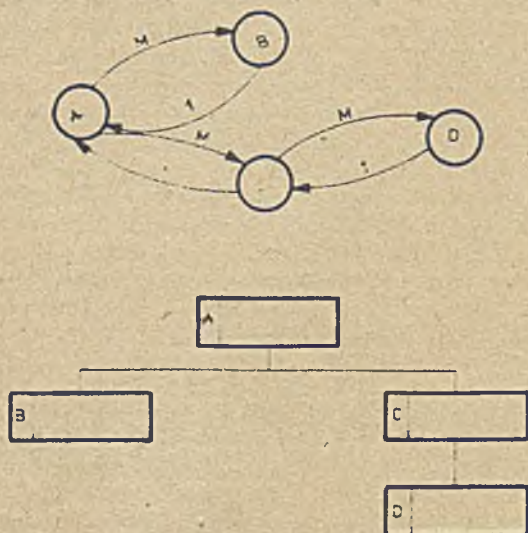
Typy asocjacji składających się na odwzorowanie nie muszą być takie same. Przykładowo asocjacja CENA-JEDNOSTKOWA: NR-CZĘŚCI, będąca odwrotną /"wstecz"/ do asocjacji typu 1 jest niemal zawsze asocjacją typu M, ponieważ wiele części może mieć tę samą cenę.

Istnieje sześć możliwych przypadków typów odwzorowań:

1 : 1	M : 1	C : 1
M : M	M : C	C : C

Ostatni z nich /C:C/ jest w normalnych warunkach sprzecznością i w praktyce nie występuje. Z pozostałych przypadków najbardziej interesujące są M : 1 i M : M między kluczami.

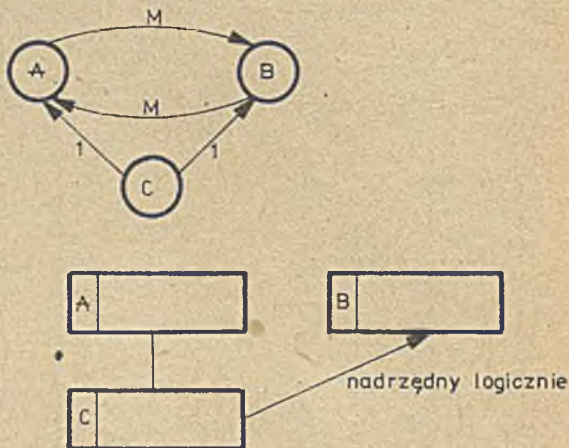
Odwzorowanie M:1 między kluczami odpowiada relacji podrzędności fizycznej między segmentami określonymi przez te klucze. Oznacza ono bowiem, że jeden z segmentów identyfikuje wiele wystąpień segmentu zależnego, zaś dla każdego segmentu zależnego istnieje dokładnie jeden segment nadrzędny /identyfikujący go/. Jest to przeciwieństwo definicji podrzędności fizycznej w systemie HADES. Jeśli segment zależny pozostaje w odwzorowaniu M : 1 z innym segmentem zależnym, trzy omawiane segmenty składają się na prosty przypadek hierarchii. Tak więc znając odwzorowania pomiędzy kluczami /i określonymi przez nie segmentami/ pakiet LSA może określić hierarchiczną strukturę logiczną między nimi.



Rys. 4. Relacje podrzędności fizycznej określone przez odwzorowanie /M:1/

Odwzorowanie M : M między kluczami pozwala pakietowi LSA na stwierdzenie relacji logicznej między nimi. Dane wystąpienie jednego z kluczy identyfikować może wiele wystąpień drugiego z nich i na odwrót. Aby taką zależność przedstawić w systemie HADES konieczne

jest zapewnienie jednoznacznego powiązania między wystąpieniami obu kluczy. Zwykle potrzebny jest trzeci element, podrzędny obu kluczom jednocześnie /tzn. identyfikujący je asocjacjami typu 1/. Ten trzeci element jest implementowany w bazach systemu HADES, jako podrzędny fizycznie jednemu z segmentów i podrzędny logicznie drugiemu z nich.



Rys. 5. Relacja podrzędności logicznej określona przez odwzorowanie /M:M/

Pakiet LSA wykrywszy pomiędzy dwoma kluczami odwzorowanie M : M, szuka segmentu podrzędnego, pozwalającego na implementację stwierdzonej relacji logicznej. Jeśli nie ma takiego segmentu, szuka on wspólnego segmentu nadrzędnego /w pewnych sytuacjach taki wspólny segment nadrzędny oddaje logikę powiązań pomiędzy segmentami w relacji logicznej, jednakże stwierdzenie czy jest tak w istocie należy do projektanta/.

Odwzorowanie C : 1 między kluczami można traktować, jako szczególny przypadek odwzorowania M : 1. Dany segment nadrzędny może mieć /nie musi/ segment podrzędny fizycznie, i jeśli go ma, to tylko jeden. Asocjacja wsteczna typu 1 gwarantuje prawidłowość implementacji tego odwzorowania w postaci hierarchii fizycznej.

Podobne względy pozwalają traktować odwzorowanie typu M : C, jako szczególny przypadek odwzorowania M : M i implementować je poprzez relację logiczną.

Odwzorowanie 1 : 1 oznacza jedną z dwu możliwości: albo pewien segment ma dwa klucze, albo między różnymi segmentami istnieje pętla /niedopuszczalna w systemie HADES/.

Wśród możliwych odwzorowań między kluczem a atrybutem na uwagę zasługuje odwzorowanie 1 : M, w którym atrybut identyfikuje wstecznie swój klucz asocjacją typu M. Oznacza to, zgodnie z definicją, że określonemu

wystąpieniu /wartości/ atrybutu odpowiada wiele wartości kluczy. Atrybut ten zostanie potraktowany przez pakiet LSA jako kandydat na indeks pomocniczy.

W pewnych przypadkach pakiet LSA będzie miał zbyt mało informacji, aby uwzględnić dane elementarne w sugerowanych przezeń strukturach logicznych. Przykładowo: nie jest on w stanie określić sposobu implementacji odwzorowań  $M : M$  między atrybutami. Sytuacja taka może pojawić się w wyniku niepełnej specyfikacji danych elementarnych na wejściu lub w wyniku błędów. Tego typu przypadki podawane są do indywidualnego rozpatrzenia przez projektanta. Przykładowo: tzw. wiszący atrybut /dana elementarna identyfikowana przez klucz asocjacją  $M/$  może być w systemie HADES implementowany jako pole powtarzalne w segmencie określonym przez klucz lub jako segment podrzędny. W tym drugim przypadku projektant musi zmienić klasyfikację danej elementarnej z atrybutu na klucz przez uzupełnienie danych wejściowych o nową daną, która stanie się atrybutem przeklasyfikowanego klucza.

#### Algorytmy przetwarzające pakietu

Realizowane przez pakiet LSA algorytmy przetwarzania danych są dość proste. Wartość pakietu polega na jego zdolności wykonywania dużej ilości prostych obliczeń i porównań z szybkością i dokładnością niemożliwymi do osiągnięcia przy działaniach manualnych. Maszyna cyfrowa analizuje wszystkie możliwe przypadki. Pakiet dokonuje następujących zalgorytmizowanych czynności:

- kontrola spójności danych wejściowych,
- klasyfikowanie danych wejściowych na klucze i atrybuty i tworzenie na tej podstawie segmentów,
- budowanie ścieżek asocjacji prostych,
- analizowanie ścieżek, wykrywanie i usuwanie asocjacji implikowanych,
- tworzenie struktur hierarchicznych,
- obliczenie wag asocjacji dla poszczególnych ścieżek,
- analizowanie kandydatów do relacji logicznych,
- określenie kandydatów do indeksowania pomocniczego.

Kontrola spójności danych wejściowych realizowana jest na dwóch poziomach. Na poziomie niższym, w ramach jednego formularza specyfikacji zapotrzebowania, bada się m. in. czy w poszczególnych polach wprowadzono wartości dowolnego typu i z dowolnych przedziałów oraz czy w części związanej z definicją danych wartości podane w polu "identyfikowany przez poziom ID" występują także w polach "poziom ID". Kontrola spójności na poziomie wyższym przeprowadzana jest między danymi wejściowymi z różnych formularzy.

Podziału na klucze i atrybuty dokonuje się w oparciu o asocjacje proste. Każdy element,

który chociaż w jednej asocjacji prostej identyfikuje inny element, jest kluczem. Pozostałe elementy są atrybutami. Klucze i atrybuty są następnie grupowane w segmenty. W skład segmentu wchodzi klucz wraz z atrybutami, które identyfikuje. Nazwa klucza staje się nazwą segmentu. Budowanie ścieżek polega na łączeniu dwóch lub więcej asocjacji prostych w łańcuch.

W ramach analizy ścieżek pakiet LSA dzieli wszystkie asocjacje występujące na ścieżkach na dwie grupy asocjacji: niezbędne i implikowane. Asocjacja niezbędna to asocjacja prosta stanowiąca jedną ścieżkę między dwoma wybranymi segmentami. Asocjacja implikowana łączy dwa segmenty, które ponadto połączone są inną ścieżką. Automatyczne usunięcie przez pakiet asocjacji implikowanej powoduje zmniejszenie redundancji tworzonej struktury logicznej bazy danych.

Efektem ubocznym analizy ścieżek jest wyznaczenie nowych asocjacji, tj. asocjacji prostych, które nie wystąpiły w danych wejściowych, lecz można je uzyskać z łączenia w ścieżkę asocjacji niezbędnych. Nowe asocjacje wskazują, jakie nowe funkcje użytkowe można utworzyć bez konieczności definiowania nowych ścieżek dostępu.

Struktury hierarchiczne powstają z kombinacji ścieżek asocjacji niezbędnych. W prostym przypadku sklejenie kilku ścieżek o wspólnych segmentach daje w wyniku hierarchię fizyczną. W przypadku skomplikowanym, gdy sklejenie ścieżek doprowadza do sytuacji, w której jeden segment na dwa segmenty nadrzędne, jeden z nich należy wybrać jako segment nadrzędny fizycznie, drugi natomiast połączyć relacją logiczną.

Wagi asocjacji stanowią względną miarę częstości użycia poszczególnych ścieżek dostępu i są często podstawowym kryterium w decyzjach determinujących parametry eksploatacyjne bazy danych. Przykładem tego typu decyzji może być problem, które segmenty umieszczać w lewej, a które w prawej ścieżce podrzędności fizycznej. Przy obliczaniu wagi asocjacji uwzględnia się takie czynniki, jak wynikowa częstość użytkowania ścieżek /iloczyn częstości dla poszczególnych elementów ścieżki dostępu/, opcja przetwarzania komend języka DL/I, czynnik on-line. Projektant może zrezygnować z obliczania wag asocjacji, jeżeli nie może uzyskać wiarygodnych estymacji częstości użycia.

Poza przypadkiem wymienionym przy omawianiu tworzenia struktur hierarchicznych, także w innych sytuacjach pary segmentów połączonych asocjacjami złożonymi lub warunkowymi kandydują do relacji logicznych. Pakiet LSA w celu wykrycia potencjalnych relacji logicznych analizuje ścieżki i poszukuje wymaganych w takich przypadkach przez SZBD HADES

danych przekroju. Dane przekroju mogą być dostarczone przez wspólny segment podrzędny wobec obu segmentów kandydujących, ewentualnie przez wspólny segment nadrzędny. Znalezienie danych przekroju nie przesądza o zaimplementowaniu związku między segmentami w postaci relacji logicznej. Decyzję dotyczącą sposobu implementacji podejmuje projektant na podstawie informacji przygotowanych przez pakiet.

#### Zasady współpracy z pakietem

Pakiet LSA został zaprojektowany z myślą o iteracyjnym wykorzystaniu pakietu, który może być uruchamiany wsadowo lub konwersacyjnie pod kontrolą podsystemu TSO. Możliwa jest także praca w trybie mieszanym, w którym wybrane fazy działania pakietu realizowane są wsadowo, inne fazy - konwersacyjnie. Pierwszy przebieg pakietu LSA wykrywa błędy wejścia, informuje o konfliktach, wskazuje sytuacje decyzyjne i sugeruje strukturę logiczną, która zwykle wymaga modyfikacji. Z punktu widzenia projektanta cenną zaletą pakietu jest wykrywanie i opisywanie wszelkich konfliktów i punktów, w których decyzja człowieka może mieć decydujący wpływ na jakość projektowanej struktury. Do ważniejszych sytuacji tego typu należą:

- asocjacje implikowane,
- nowe asocjacje,
- synonimy i homonimy,
- asocjacje niespójne,
- pętle i ścieżki zbyt długie,
- wybór segmentu nadrzędnego flizycznie,
- relacje logiczne.

Najogólniej, w pełnym przebiegu pakietu można wydzielić dwie fazy:

- diagnostyczną / rozumianą bardzo szeroko/,
- projektową.

przy czym faza diagnostyczna nie jest niepodzielna całością; w iteracyjnym uruchamianiu pakietu kolejne wywołania fazy diagnostycznej nie zawsze muszą rozpoczynać się od początku fazy. Po przeanalizowaniu raportów diagnostycznych użytkownik decyduje, od którego punktu należy wznowić przerwane przetwarzanie, które fazy lub ich fragmenty należy powtórzyć. Projektant poprawia ewentualne błędy, nanosi zmiany i swoje decyzje komunikuje pakietowi w postaci komend specyfikacji struktury. Wybrane przykładowo komendy specyfikacji struktury mają na celu zilustrowanie możliwości dostarczanych przez to narzędzie.

- USUŃ - komenda usuwa wszystkie asocjacje pomiędzy wskazanymi elementami danych,
- KONIECZNA - komenda włącza asocjację implikowaną do struktury logicznej, jako asocjację niezbędną,
- DOMINUJĄCY - komenda wybiera segment nadrzędny fizycznie /w sytuacji, gdy istnieją dwa segmenty nadrzędne/,
- PODRZĘDNY - komenda ustanawia między dwoma elementami relację podrzędności fizycznej,
- KLUCZ-SEGM - komenda zmienia nazwę segmentu, ustala pozycję klucza,

ID-KWANT - komenda wskazuje element, który zostanie atrybutem określonego klucza.

Projektant ma do dyspozycji także komendy modyfikacji wejścia, które pozwalają dokonywać określonych zmian w danych wejściowych bez konieczności ich ponownego wprowadzania. Komendy modyfikacji wejścia mają następujące znaczenie:

- GRUPA - pozwala na połączenie dwóch segmentów w jeden przez konkatencję ich kluczy,
- POMINDEKS - wskazuje element, który będzie indeksem pomocniczym,
- SYNONIM - rozwiązuje problem synonimów przez wskazanie, że dane nazwy znaczą, są synonimami,
- IDENTYCZNOŚĆ - rozwiązuje sytuacje identyczności, przez wskazanie elementu, który ma zostać kluczem /drugi element będzie atrybutem/.

Projektant modyfikuje dane wejściowe i struktury utworzone przez pakiet po przeanalizowaniu raportów diagnostycznych i projektowych. Może też modyfikować struktury sugerowane przez pakiet LSA bez żadnej sygnalizacji błędów, proponując swoje rozwiązania lub warianty i zlecając pakietowi ich przeanalizowanie. Drugi i ewentualnie kolejne przebiegi uwzględniają modyfikacje wprowadzone za pomocą komend. Każda z komend obowiązuje w następnych iteracjach i nie musi być powtarzana. Projektant może też anulować lub zmienić wcześniej wprowadzoną komendę. Proces iteracyjnej współpracy z pakietem LSA zostaje zakończony, gdy projektant uzna, że przedstawiony projekt w pełni spełnia wymagania przyszłych użytkowników bazy danych.

Raporty projektowe stanowią rezultat działania pakietu LSA i są dokumentacją struktury logicznej projektowanej bazy. W fazie projektowej tworzone są następujące raporty:

1. Sugerowane segmenty. Lista wszystkich atrybutów zgrupowanych wokół określonego klucza. Podaje nazwę klucza, nazwy atrybutów i długość segmentu.
2. Model strukturalny. Lista kluczy i wszystkich ich asocjacji uporządkowanych hierarchicznie. Dla kluczy, które mogą mieć więcej niż jeden segment nadrzędny podawane są nazwy alternatywnych segmentów nadrzędnych /kandydatów do relacji logicznych/. Lista zawiera podsumowanie wag asocjacji.
3. Graf struktur fizycznych. Ostateczny kształt sugerowanej przez pakiet hierarchii fizycznej.
4. Kandydaci do indeksowania pomocniczego. Lista kluczy typowanych do indeksowania pomocniczego przez pakiet oraz elementów danych wskazanych przez komendy POMINDEKS lub IDENTYCZNOŚĆ.

Należy także wspomnieć o dodatkowym wyjściu pakietu LSA dostosowanym do wejścia innego pakietu, wchodzącego w skład oprogramowania pomocniczego systemu zarządzania

bazą danych HADES, pakietu HDSMAP wspomagającego projektowanie struktury fizycznej. Po utworzeniu struktury logicznej projektowanej bazy danych i zaakceptowaniu jej przez projektanta, pakiet LSA generuje szkielety opisów fizycznych baz danych /DBD/, odpowiadających strukturze logicznej proponowanej bazy danych.

#### Przykład modelowania struktury logicznej

Walory użytkowe pakietu LSA zostaną przedstawione na przykładzie dotyczącym bazy danych dla dziekanatu wyższej uczelni. Adresatami działalności dziekanatu są studenci. Dziekanat posiada informacje o przedmiotach prowadzonych w ramach danego wydziału. Przed rozpoczęciem semestru wykładowcom dostarczane są listy grup studenckich. Dziekanat przed końcem semestru przygotowuje karty zaliczeń dla poszczególnych przedmiotów i grup studenckich. Następnie, na podstawie wyników zaliczeń, sporządzane są protokoły egzaminów. Listy ocen uzyskanych przez studentów są podstawą do zaliczenia semestru, przygotowania protokołów poprawkowych, a także uzyskania statystyk ocen ilustrujących przebieg studiów indywidualnych studentów czy grup studenckich.

Baza danych dziekanatu zawiera informacje o studentach, pracownikach dydaktycznych, grupach przedmiotowych, uzyskanych ocenach. Dane z bazy są wykorzystywane w siedmiu programach użytkowych, których krótki opis podano niżej.

1. Grupy przedmiotowe. Funkcja realizuje zapytania dotyczące przedmiotów prowadzonych na danym wydziale uczelni wyższej.
2. Weterani. Funkcja dostarcza informacji o poszczególnych grupach studenckich i studentach, sporządza listę studentów z określonej grupy.
3. Wystawianie ocen. Funkcja na żądanie dostarcza informacji o ocenach ze wskazanego przedmiotu. Podaje się dane dotyczące każdego studenta: nazwisko, numer legitymacji i uzyskaną ocenę.
4. Samorząd studencki. Funkcja realizuje zapytania dotyczące samorządu studenckiego, jego agend, funkcji pełnionych w samorządzie itp.,
5. Karty zaliczeń. Funkcja generuje karty zaliczeń wg grup studenckich, przedmiotów i prowadzących zajęcia dydaktyczne.
6. Protokoły poprawkowe. Funkcja dostarcza dla określonych grup studenckich protokoły po-

prawkowe. W protokóle umieszczane są nazwiska studentów, nazwa przedmiotu, nazwiska egzaminujących.

7. Statystyka ocen. Funkcja dostarcza danych statystycznych o ocenach uzyskanych przez studentów w okresie studiów.

Analiza każdej z wymienionych funkcji użytkowych została przeprowadzona niezależnie od innych. W rezultacie uzyskano siedem dokumentów wejściowych do pakietu LSA. W celu uzyskania większej przejrzystości, na standardowych formularzach zakodowano wyłącznie informacje niezbędne dla poprawnej pracy pakietu.

#### Przykłady raportów końcowych

Efektom końcowym procesu projektowania struktury logicznej bazy danych są raporty projektowe, a zwłaszcza raporty przedstawiające sugerowane segmenty, graf struktur fizycznych oraz kandydatów do indeksowania pomocniczego. W podanym wyżej przykładzie modelowania, dotyczącym projektowania bazy danych dla dziekanatu, zamieszczono przetestowane wcześniej dane wejściowe. Otrzymane raporty diagnostyczne wykazały spójność danych wejściowych. W tym nietypowym z punktu widzenia praktyki projektowej przebiegu pakietu LSA iteracja okazała się zbędna /w rzeczywistości należy przyjąć, że było kilka iteracji, ograniczono się jednak do przedstawienia wyników ostatecznych.

#### L i t e r a t u r a :

- [1]. C. J. Date, Wprowadzenie do baz danych, WNT, Warszawa, 1981.
- [2]. System Zarządzania Hierarchiczną Bazą Danych - HADES. Wprowadzenie. Oprogramowanie OS/JS, ZE ELWRO Wrocław, 1984.
- [3]. T. Mc Elreath: IMS Design and Implementation Techniques, Q. E. D. Information Sciences, Wellesley, 1981.
- [4]. N. Raver, G. Hubbard: Automated Logical Data Base, Design: Concepts and Applications, W: IBM Systems Journal nr 3/1977, s. 287-312.
- [5]. K. Sycz, R. Nikodem: Projektowanie struktur logicznych baz danych z wykorzystaniem narzędzi komputerowych, Materiały INFOGRYF'84, tom 2, s. 177-185.
- [6]. K. Sycz, R. Nikodem, N. Tarczyńska: Wspomaganie projektowania logicznych struktur hierarchicznych baz danych, Materiały Konferencji "Komputerowe wspomaganie projektowania", Politechnika Poznańska, 1983.

Potrzeba szybkiego selektywnego dostępu do danych zapisanych w pamięci zewnętrznej komputera jest jedną z przesłanek organizacji wspólnych baz danych. Typowymi przykładami są bazy: Zakładu Ubezpieczeń Społecznych, bibliotek, central handlowych i hurtowni oraz szpitali. Systemy informacyjne tych przedsiębiorstw lub instytucji powinny umożliwiać pracę konwersacyjną oraz bezpośredni, prawie natychmiastowy, dostęp do pojedynczych rekordów lub segmentów. Funkcjonujące w systemie zarządzania hierarchiczną bazą danych HADES /zwanym dalej systemem HADES/ metody HDAM i HIDAM umożliwiają bezpośredni dostęp do pojedynczych rekordów. Metoda HIDAM wymaga przed utworzeniem bazy uporządkowania rekordów i w związku z tym nadaje się do baz o ustabilizowanej strukturze i liczebności, o małym współczynniku dynamiki rekordów. Ograniczeń tych nie posiada druga z metod bezpośrednich, tj. metoda HDAM.

Generalnie proces randomizacji kluczy sprowadza się do równomiernego odwzorowania zbioru wartości kluczy /który jest potencjalnie bardzo liczny, np. numery części samochodowych/ w numery bloków bazy, których liczba wynika z aktualnej liczebności typów części i jest wielokrotnie mniejsza od potencjalnej liczby kluczy. Dotychczas nie opracowano metody gwarantującej równomierny rozkład rekordów w blokach dla dowolnego rozkładu wartości atrybutów wyszukiwania, tj. kluczy. Istnieją wprawdzie opracowania o charakterze poradnikowym [7], [1] ale są one trudno dostępne i nie dotyczą organizacji baz typu HDAM lecz pewnej uogólnionej organizacji modelowej.

Celem niniejszego artykułu jest prezentacja założeń konstrukcyjno-technologicznych pakietu programowego wspomagającego pracę projektanta i programisty użytkowego, oraz administratora bazy danych w zakresie prawidłowego tworzenia i eksploatacji baz danych typu HDAM. Pakiet, któremu nadano nazwę HDSRAND, jest zestawem interakcyjnych programów eksploatowanych pod kontrolą systemu OS/JS z opcją TSO na komputerze R-32 z minimalną pamięcią operacyjną 512 KB.

Główne funkcje pakietu HDSRAND są następujące:

- przygotowanie zbioru wartości identyfikatora /kluczy/, na podstawie których będzie realizowany proces zakładania bazy,
- zebranie parametrów istniejącej lub modelowanej bazy oraz urządzeń pamięci dyskowej, na której planuje się jej przechowywanie,
- wybór procedury randomizacyjnej lub włą-

czenie procedury dostarczonej przez użytkownika,

- symulacja działania wybranego algorytmu randomizacji,
- generowanie raportów z przebiegu modelowania,
- aktualizacja wybranych parametrów modelowanej bazy danych, charakterystyk kluczy lub zmiana procedury randomizacyjnej.

Pakiet HDSRAND jest fragmentem interaktywnego laboratorium projektowania baz danych LABADA [8]. Jednocześnie może on być eksploatowany jako pakiet samodzielny u użytkowników, którzy nie posiadają laboratorium LABADA. W drugim przypadku pakiet zostanie uzupełniony o te programy i dane laboratorium LABADA, które są niezbędne dla realizacji jego funkcji. Wygenerowanie odpowiedniej wersji będzie realizowane przez dostawcę pakietu w momencie przygotowania taśmy dystrybucyjnej.

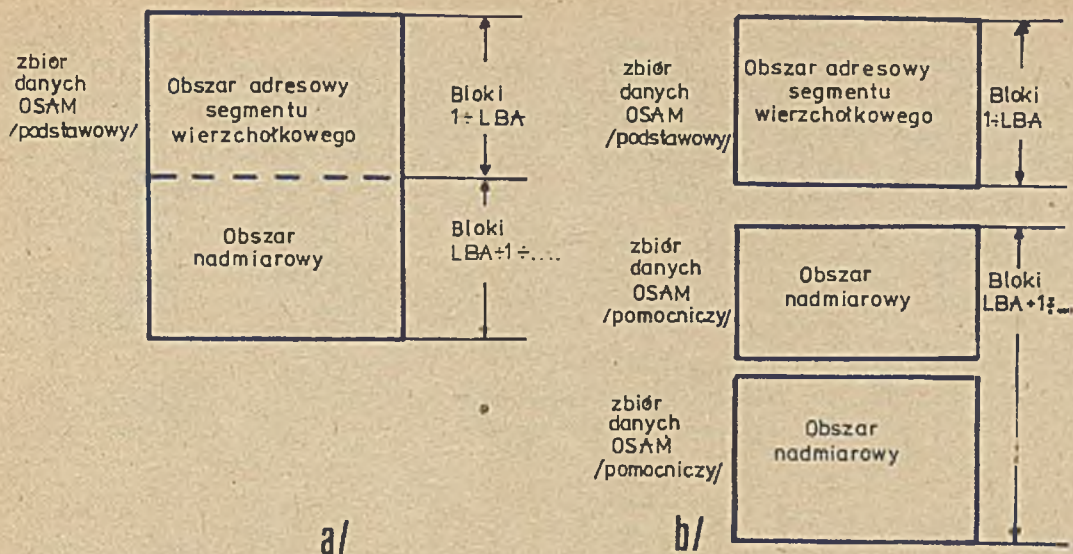
Zainstalowanie pakietu przynosi wiele korzyści, takich jak:

- dobór najwłaściwszej z dostępnych procedur randomizacyjnych dla określonego zbioru kluczy, przy czym klucze mogą być automatycznie pobierane z istniejących zbiorów danych, mogą stanowić reprezentacyjną próbę lub zostać automatycznie wygenerowane za pomocą zawartych w pakiecie generatorów kluczy o dowolnej budowie,
- poprawę parametrów eksploatacyjnych /głównie czasu dostępu/ do istniejących baz typu HDAM,
- oszacowanie oczekiwanej liczby prób dostępu do rekordu przy zadanych parametrach wielkości bloku, liczby punktów zakotwiczenia oraz wielkości /liczby bloków/ podstawowego i nadmiarowego obszaru zbioru OSAM,
- rozbudowę istniejącego zbioru procedur randomizacyjnych o co najmniej trzy dalsze procedury randomizacyjne oraz
- skrócenie czasu zakładania bazy typu HDAM przez właściwe "zestrojenie" parametrów alokacyjnych.

#### Charakterystyka baz danych HDAM

Na bazę danych typu HDAM składa się jeden podstawowy zbiór danych OSAM oraz od zera do dziewięciu pomocniczych zbiorów danych /również OSAM/. Strukturę dwóch przykładowych baz HDAM ilustruje rys. 1. Obszar adresowy segmentu wierzchołkowego umieszcza się tylko w podstawowym zbiorze danych. Wszystkie pozostałe stanowią obszar nadmiarowy. Podstawową różnicą pomiędzy strukturą HDAM





Rys. 1. Struktura baz danych HDAM: a - jeden podstawowy zbiór danych, b - jeden podstawowy oraz dwa pomocnicze zbiory danych

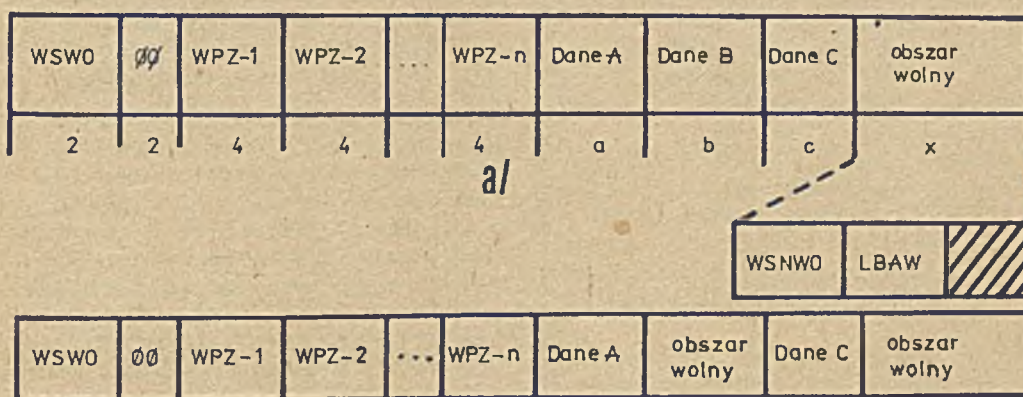
a trzema pozostałymi /HSAM, HISAM, HIDAM/ jest to, iż początkowe ładowanie bazy danych nie musi być wykonywane w określonym porządku segmentów wierzchołkowych. Natomiast segmenty podrzędne powinny być podawane w porządku hierarchicznym przed ładowaniem kolejnego segmentu wierzchołkowego. Każdy blok w bazie danych HDAM zawiera, oprócz wskaźników wolnego obszaru w bloku, wierzchołkowe punkty zakotwiczenia /rys. 2/.

W celu zapisania segmentu wierzchołkowego do bazy HDAM, procedura randomizująca wywołana z programu użytkowego przekształca klucz tego segmentu na adres złożony z nume-

ru bloku oraz numeru punktu zakotwiczenia. Jeśli kolejne segmenty wierzchołkowe są adresowane jednakowo /stanowią synonimy/, to wówczas wierzchołkowy punkt zakotwiczenia wskazuje na segment wierzchołkowy z najmniejszą wartością klucza, natomiast wszystkie kolejne synonimy są ze sobą połączone z użyciem wskaźników PHYSICAL TWIN w sekwencji wartości klucza.

#### Charakterystyka procesu modelowania

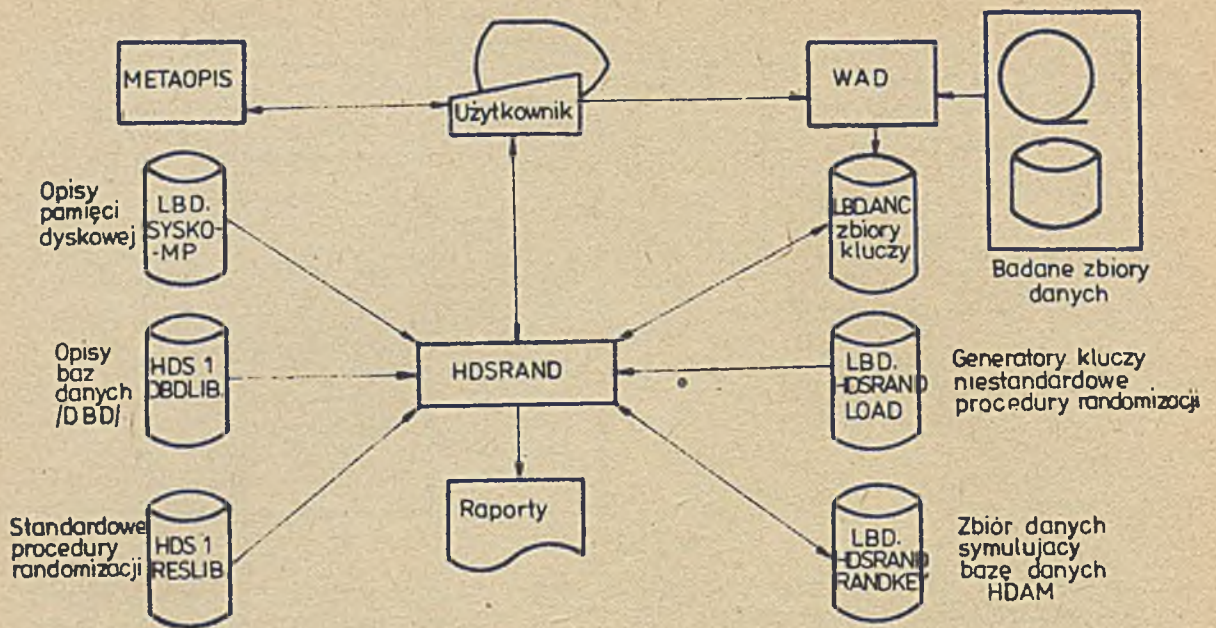
Proces modelowania bazy o organizacji HDAM realizowany jest w kilku etapach i biorą w nim udział zbiory danych /sekwencyjne, bezpośrednie oraz biblioteczne/ przedstawio-



#### Oznaczenia:

- WSWO - wskaźnik wolnego obszaru
- WPZ-x - wierzchołkowy punkt zakotwiczenia-x
- WSNW0 - wskaźnik następnego wolnego obszaru
- LBAW - liczba bajtów obszaru wolnego

Rys. 2. Struktura bloku HDAM: a - po załadowaniu segmentów A, B, C, b - po usunięciu segmentu B



Rys. 3. Ogólny schemat przetwarzania programu HDSRAND

ne na rys. 3. Użytkownik uruchamia pakiet HDSRAND za pomocą procedury komend TSO z parametrem określającym nazwę biblioteki przechowującej skompilowane opisy DBD /wartością domyślną jest HDS1, DBDLIB/. Dalsze sterowanie pracą zawarte jest w programach pakietu. Wymienione na rys. 3 zbiory pełnią następujące funkcje:

- LBD. SYSKOMP - zawiera opisy modelowanych urządzeń pamięci zewnętrznych komputerów JS EMC. Jego zawartość może być zmieniana i uzupełniana za pomocą programu METAOPIS laboratorium LABADA, również w sposób konwersacyjny.
- HDS1. DBDLIB - zawiera skompilowane opisy DBD. Jeżeli użytkownik posiada własną bibliotekę, to musi podać jej nazwę w momencie rozpoczynania pracy z pakietem.
- HDS1. RESLIB - przechowuje standardowe procedury randomizacyjne dla systemu HADES.
- Id. TSO. LBD. ANC - prywatny zbiór użytkownika TSO, w którym może przechowywać klucze pozyskane automatycznie za pomocą procedury AN CZEST programu WAD laboratorium LABADA.
- LBD. HDSRAND. LOAD - biblioteka zawierająca procedury generowania kluczy oraz własne procedury randomizacyjne użytkownika.
- LBD. HDSRAND. RANDKEY - zbiór danych symulujący modelowaną bazę HDAM. Jego standardowa wielkość umożliwia modelowanie baz do 200 000 rekordów i może być rozszerzony przez użytkownika.

Zanim użytkownik przystąpi do współpracy z pakietem HDSRAND, powinien podjąć decyzję o tym, czy zbiór kluczy podlegających ran-

domizacji zostanie utworzony za pomocą programu WAD oraz czy opis typu pamięci dyskowej, w której będzie alokowana baza danych został wprowadzony do metabazy laboratorium /do zbioru LBD. SYSKOMP/. Jeśli zdecydował się na badania istniejącego zbioru za pomocą programu WAD, to powinien wywołać ten program za pomocą procedury komend:

wad nazwa-badanego-zbioru-danych  
numer-serijny-woluminu

Także w przypadku, gdy w zbiorze LBD. SYSKOMP nie istnieje żądany opis pamięci dyskowej /można to sprawdzić za pomocą programu PREZOPIS/, należy wywołać program METAOPIS w celu wprowadzenia tego opisu.

Kolejne czynności przygotowawcze mogą polegać na włączeniu do biblioteki LBD. HDSRAND. LOAD własnych procedur randomizujących lub generujących klucze<sup>1/</sup>. HDSRAND pracuje w trybie interaktywnym prowadząc z użytkownikiem dialog o charakterze inwazyjnym. Dzięki temu z jego usług mogą korzystać użytkownicy nieprzeszkoleni, nie posiadający doświadczenia w stosowaniu systemu TSO. Wymagana jest jedynie merytoryczna znajomość niektórych problemów projektowania hierarchicznych baz danych.

Proces modelowania odbywa się w czterech podstawowych etapach:

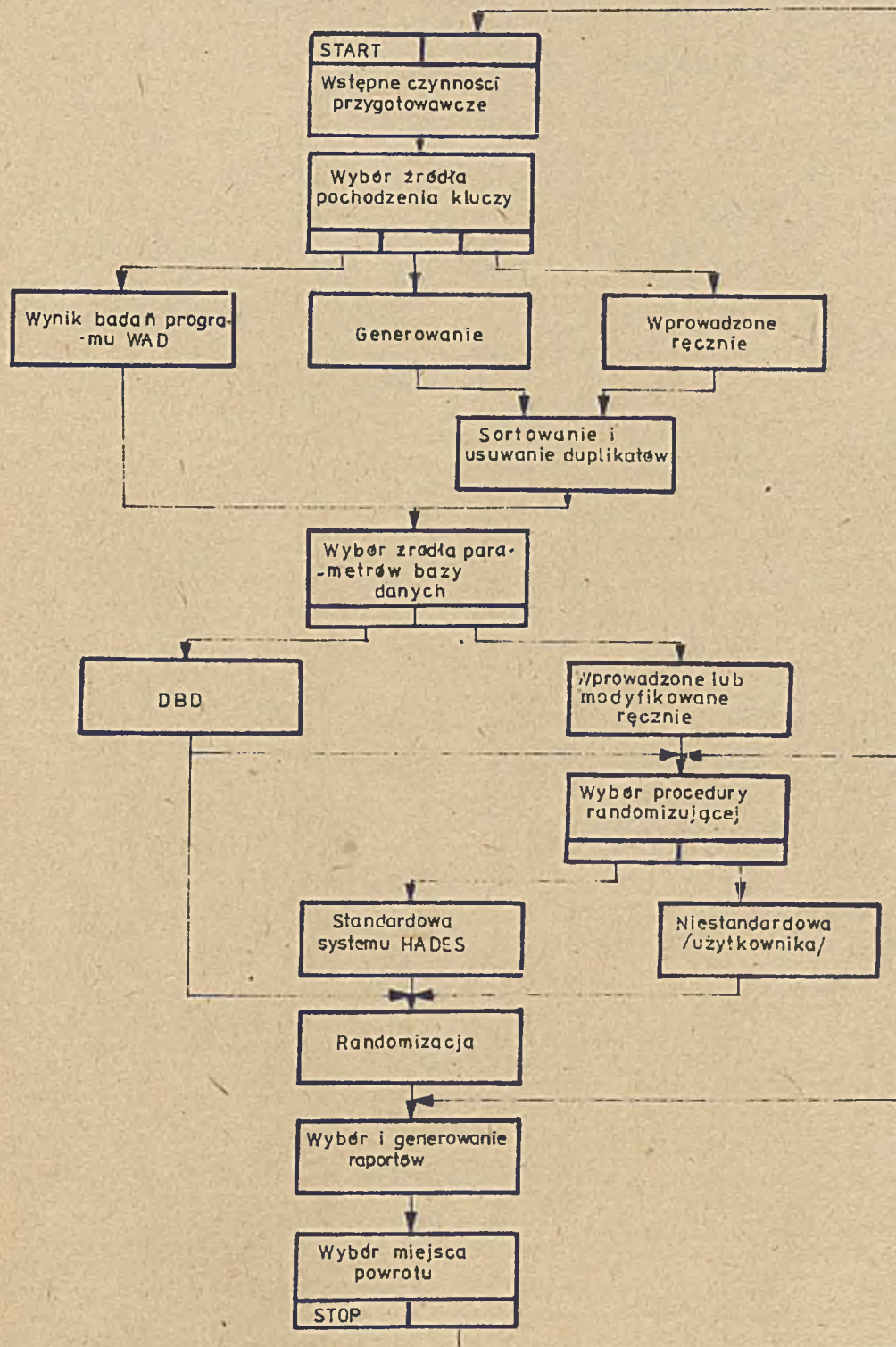
<sup>1/</sup> Pakiet HDSRAND w wersji dystrybucyjnej zawiera generatory kluczy posiadających wszystkie typowe rozkłady oraz typowe procedury randomizacyjne.

1. dostarczenie zbioru kluczy podlegających randomizacji,
2. dostarczenie parametrów bazy danych,
3. symulacja randomizacji,
4. wybór i generowanie raportów - rys. 4.

W każdym z wymienionych etapów /za wyjątkiem randomizacji/ użytkownik otrzymuje do wyboru kilka wariantów dalszej pracy. Po za-

kończeniu procesu modelowania istnieje możliwość powtórzenia wybranych etapów pracy w zmienionych warunkach /inny rozkład kluczy, inne parametry bazy danych lub procedura randomizująca/.

Przygotowanie danych do modelowania  
Pierwsze dwa z wymienionych etapów modelowania dostarczają danych niezbędnych do u-



Rys. 4. Ogólny schemat przebiegu modelowania procesów randomizacji za pomocą programu HDSRAND

ruchomienia procesu randomizacji oraz do jego prawidłowej interpretacji. Pierwszą czynnością jest udostępnienie odpowiedniego zbioru kluczy do przetwarzania. Możliwy jest wybór jednego z trzech wariantów:

1. Skorzystanie z wyników badań programu WAD zawartych w zbiorze LBD.ANC.
2. Wygenerowanie /za pomocą jednego z dostępnych w pakiecie generatorów/ zbioru kluczy o żądanych cechach, takich jak: liczebność, długość, typ i rozkład.
3. Ręczne wprowadzanie kluczy w trakcie pracy programu HDSRAND.

Jeżeli użytkownik wybierze wariant drugi lub trzeci, otrzyma również możliwość posortowania kluczy oraz usunięcia duplikatów, a także uzupełnienia zbioru o dowolną liczbę nowych kluczy<sup>2/</sup>. Standardowo dostępnymi generatorami kluczy są procedury dające rozkłady: jednostajny, skokowy, normalny, Poissona, modulowany-1 /sinusoidalny/, modulowany-2 /wyznaczony przez wielomian/. Drugi etap pracy programu HDSRAND ma dostarczyć informacji o modelowanej bazie danych, typu HDAM. Źródłem tych danych może być istniejący /skompilowany/ opis w postaci DBD, zawarty w bibliotece HDS1, DBDLIB lub opis wprowadzony ręcznie w trakcie pracy programu HDSRAND.

Opis wczytany z biblioteki HDS1, DBDLIB może być przeglądany, uzupełniany i modyfikowany. Program HDSRAND dba o zachowanie spójności i zgodności danych. Parametrami kompletowanymi na tym etapie pracy programu są:

- liczba ładowanych segmentów wierzchołkowych,
- liczba bloków obszaru adresowego segmentu wierzchołkowego,
- przeciętna liczba segmentów wierzchołkowych w bloku x obszaru adresowego,
- liczba punktów zakotwiczenia w bloku,
- sposób rozmieszczenia wolnego obszaru w bazie,
- model pamięci dyskowej, w której będzie alokowana baza danych<sup>3/</sup>,
- nazwa procedury randomizującej.

Procedury randomizujące mogą znajdować się w bibliotece LBD, HDSRAND, LOAD. W drugim przypadku mogą to być procedury dostarczane wraz z pakietem HDSRAND lub napisane przez użytkownika.

#### Procedury randomizacyjne

System HADES dostarcza użytkownikowi trzy standardowe procedury randomizacji:

1. moduło lub dzielenia /DFSHDC10/.
2. binarnego połówkowania /DFSHDC20/.
3. haszowa /DFSHDC30/.

Trzy procedury randomizacyjne, zawarte w systemie HADES, proponuje się uzupełnić o trzy procedury dodatkowe. Uwzględniają one stały postęp dokonujący się w zakresie badań nad randomizatorami oraz mogą eli-

minować pewne niedogodności związane ze stosowaniem procedur standardowych.

Metoda dzielenia przez liczbę pierwszą stanowi modyfikację wymienionej wcześniej metody dzielenia. Jeżeli dzielnikiem będzie liczba pierwsza mniejsza od ogólnej liczby punktów zakotwiczenia to możemy oczekiwać, że nie dojdzie do grupowania się segmentów wokół adresów parzystych. Metodę tę określamy symbolem DIVPRN /ang. division by prime number/.

Metoda mnożenia [3], s. 18-19/ zwana również funkcją mieszającą Fibonacciego zakłada mnożenie klucza przez stałą

$$c = \sqrt{5-1}/2 \approx 0.61803398875$$

Zatem adres o postaci względnego numeru punktu zakotwiczenia w bazie otrzymamy po wykonaniu:

$$[LPUNZAK * //c + klucz/modi//]$$

gdzie:

[x] oznacza liczbę całkowitą nie większą od  $x$   
 $x/modi$  jest częścią ułamkową liczby  $x$

Dzieląc względny numer punktu zakotwiczenia w bazie przez liczbę punktów zakotwiczenia w bloku otrzymujemy względny numer bloku w bazie. Reszta z dzielenia stanowi numer punktu zakotwiczenia w bloku. Ostateczny adres otrzymujemy po zwiększeniu otrzymanych wyników o jedynek.

Metoda analizy znaków zależna jest od charakterystyki kluczy /poprzednie metody nie były zależne/. Dzięki przebadaniu zbioru kluczy za pomocą procedury ANZNAK programu WAD możemy otrzymać listę pozycji klucza uporządkowaną według równomierności występowania znaków. Do typowania pozycji znakowych klucza została zastosowana metoda zaproponowana przez J. Korczaka [5]. Algorytm tej metody wbudowany do programu prezentacji analizy struktury zbioru - PREZWAD, uwalnia projektanta od przeprowadzania żmudnych obliczeń. Zadaniem projektanta jest jedynie wskazanie, ile pozycji znakowych klucza będzie argumentem działania procedury randomizacyjnej. Symbolem tej metody jest CHARAN /ang. characters analysis/.

#### Analiza i prezentacja wyników modelowania

W wyniku wykonania randomizacji otrzymuje się zbiór LBD, RANDKEY symulujący bazę danych HDAM. Zbiór ten, wraz z opisem bazy danych, stanowi podstawę do dalszych analiz.

Ostatnim etapem pracy programu HDSRAND jest analiza i prezentacja wyników modelowa-

<sup>2/</sup> Klucze w zbiorze LBD.ANC po badaniu programem WAD są posortowane.

<sup>3/</sup> Wszystkie niezbędne charakterystyki pamięci dyskowej zostaną automatycznie odczytane ze zbioru LBD, SYSKOMP.

nia. W jej wyniku użytkownik może otrzymać sześć rodzajów raportów:

- 1/ analiza statystyczna przebiegu modelowania,
- 2/ rozkład liczby adresowanych segmentów wierzchołkowych do bloków w obszarze adresowym,
- 3/ rozkład długości łańcuchów randomizacji,
- 4/ wykaz bloków z podaną liczbą adresowanych segmentów wierzchołkowych,
- 5/ wykaz adresów z podaną liczbę adresowanych segmentów wierzchołkowych,
- 6/ wykaz wartości kluczy wraz z wygenerowanymi adresami,

Pierwszy z wymienionych raportów sporządzany jest obowiązkowo i stanowi najbardziej ogólny i zarazem syntetyczny przegląd wyników modelowania. Zawiera, obok danych wejściowych, następujące charakterystyki:

- współczynnik wypełnienia pamięci w obszarze adresowym segmentów wierzchołkowych,
- liczbę bloków przepełnionych,
- przeciętną liczbę adresowanych segmentów wierzchołkowych do jednego bloku,
- przeciętną liczbęostępów w celu wyszukania segmentu wierzchołkowego,
- odchylenie standardowe liczbyostępów,
- liczbę segmentów wierzchołkowych w obszarze nadmiarowym,
- przeciętną długość łańcucha randomizacji,
- odchylenie standardowe długości łańcucha randomizacji,
- przeciętną długość łańcucha w obszarze nadmiarowym.

Zawartość tego raportu pozwala użytkownikowi dokonać pierwszej oceny przydatności modułu randomizacji do konkretnego zestawu wartości klucza oraz parametrów opisu bazy danych. Podstawowym parametrem decydującym o ocenie jest przeciętna liczbaostępów wykonywanych w celu wyszukania segmentu wierzchołkowego. Pozostałe parametry mogą służyć do wstępnej diagnostyki w sytuacji, gdy otrzymano wyniki niezadowolające /zbyt małe wypełnienie pamięci, zbyt duży narzut z powodu dodatkowychostępów itp. / . Stosunek liczby bloków przepełnionych do ogólnej liczby bloków w obszarze adresowym, w porównaniu ze współczynnikiem wypełnienia pamięci, pozwoli wstępnie ocenić równomierność rozkładu adresów wyliczonych przez procedurę randomizującą. Natomiast informacje o długości łańcucha randomizacji mają na celu ocenę prawidłowości wyboru liczby punktów zakotwiczenia w bloku. Wszystkie następne rodzaje raportów stanowią zestawienia o rosnącej skali szczegółowości oraz o rosnącej objętości. Dzięki tym raportom można dokładniej niż na podstawie zestawienia podstawowego przeanalizować modelowane decyzje projektowe.

Rozkłady: liczby adresowych segmentów wierzchołkowych oraz długości łańcuchów randomizacji są zestawieniami o podobnej strukturze /tabela 1/. Podane są szeregi rozdzielcze liczebności oraz częstości /w procentach/, a także szereg skumulowany częstości.

Tabela 1

Rozkład liczby adresowanych przez procedurę FIBO segmentów wierzchołkowych do bloków w obszarze adresowym bazy BAZASTR

Liczba adresowanych segmentów	Liczba bloków	%	Suma %
1	2	3	4
0	0	0,00	0,00
1	2	0,06	0,06
2	86	2,69	2,75
3	734	23,03	25,78
4	855	26,72	42,50
5	1327	41,47	83,97
6	22	0,68	84,65
Razem	3200	100,00	

Rozkład liczby adresowanych segmentów wierzchołkowych może być wykorzystany do oceny poprawności działania wybranej procedury randomizującej. Otrzymany rozkład powinien posiadać własność: rozkładu normalnego o wartości średniej równej pojemności bloku /liczonej w segmentach wierzchołkowych/ pomnożonej przez współczynnik wypełnienia pamięci w obszarze adresowym segmentów wierzchołkowych i odchyleniu standardowym jak najmniejszym. Wszelkie odchylenia od pożądanego kształtu rozkładu /na przykład dwa lub więcej ekstrema/ świadczą o błędnym wyborze procedury randomizującej. Podobne uwagi można odnieść do rozkładu długości łańcuchów randomizacji. Jeżeli otrzymany w wyniku modelowania rozkład posiada jedno ekstre-

Tabela 2

Wykaz bloków oraz adresów w bazie BAZASTR wypełnianych segmentami przez procedurę FIBO

Numer bloku	Numer punktu zakotwiczenia	Liczba adresowanych segmentów wierzchołkowych	Liczba segmentów przepełnienia
1	2	3	4
1		8	3
	1	3	
2	2	5	
		4	∅
2	1	2	∅
	2	2	
3		2	∅
	1	∅	
	2	2	

Wykaz wartości kluczy oraz adresów  
wygenerowanych przez procedurę  
FIBO dla bazy BAZASTR

Numer klucza	Wartość klucza	Numer bloku	Numer punktu zakotwiczenia	Numer klucza	Wartość klucza
1	2	3	4	1	2
1	1002784	728	1	51	183
2	1003845	3272	2	52	183
3	1003846	273	1	53	183
4	1003847	984	1	54	184
5	1022777	1724	1	55	190
6	1022778	487	2	56	1

mum, wartość średnią równą ilorazowi liczby ładowalnych do bazy segmentów wierzchołkowych przez liczbę możliwych do wygenerowania różnych adresów i małe odchylenie standardowe, to znaczy, że wybrany moduł randomizacyjny jest dobry.

Kolejne dwa raporty zawierają wykazy: bloków wraz z podaną dla każdego bloku liczbą adresowanych do niego segmentów wierzchołkowych, oraz adresów wraz z podaną liczbą synonimów /będące jednocześnie długością łańcucha randomizacji/. Ponieważ obydwa raporty posiadają podobną strukturę, istnieje możliwość ich połączenia w jedno zestawienie /tabela 2/.

Raport ten pozwala, w przypadku braku pozytywnej oceny wyników modelowania, zlokalizować miejsca szczególnie licznych kolizji. Takie działanie powinno umożliwić /na podstawie omawianego raportu oraz raportu ostatniego, tzn. wykazu wartości kluczy i wygenerowanych adresów/ poprawienie badanej procedury randomizującej lub wybór innej, bardziej przystosowanej do badanego zbioru kluczy. Ostatni raport jest najobszerniejszy i zawiera wykaz wszystkich kluczy wraz z wygenerowanym przez procedurę randomizacji adresem. Strukturę tego raportu przedstawiono w tabeli 3. Umożliwia on szczegółową analizę działania procedury randomizującej i /w przypadku działania niezgodnego z założeniem/ jej poprawę. Raport ten będzie miał szczególnie duże znaczenie do testowania nowych procedur randomizacji tworzonych przez użytkownika.

#### Zastosowanie pakietu HDSRAND w procesie projektowania hierarchicznych baz danych

Zastosowanie pakietu HDSRAND może być dwójakie. Z jednej strony może on stanowić narzędzie projektanta i programisty w celu określenia parametrów bazy HDAM i wyboru najwłaściwszej /dla określonego klucza/ procedury randomizacyjnej. Uzyskane wyniki tej

fazy modelowania organizacji bazy HDAM mogą stanowić następnie wejście dla programu HDSMAP, który umożliwi dokładniejsze oszacowanie wielkości bazy oraz przedstawi wielkości narzutów obszaru systemowego oraz technologicznego. Z drugiej strony pakiet HDSRAND może stanowić narzędzie administratora bazy danych w celu poprawy funkcjonowania założonych już baz typu HDAM. Może on służyć przede wszystkim symulowanej zmianie głównych parametrów bazy, tj. współczynnika wypełnienia oraz długości bloku i liczby punktów zakotwiczenia.

#### L i t e r a t u r a :

- [1] W. Bucholz: File Organization and Addressing. IBM System: Journal, June 1963, s. 86-111.
- [2] System zarządzania bazą danych HADES. Podręcznik programisty systemowego. ŻE ELWRO, Wrocław 1983.
- [3] R. Jagielski: Tablice rozproszone. WNT Warszawa 1982.
- [4] G. D. Knott: Hashing Functions. The Computer Journal, vol. 18, s. 265-278, 1975.
- [5] J. J. Korczak: Metoda analizy identyfikatorów zapisów w zbiorach o organizacji zrandomizowanej. Prace Naukowe Akademii Ekonomicznej we Wrocławiu, nr 102/1977, s. 137-152.
- [6] J. J. Korczak: Problemy budowy optymalnych struktur danych w pamięciach dyskowych. Praca doktorska, Akademia Ekonomiczna we Wrocławiu 1978 /maszynopis/.
- [7] V. Y. Lum: General Performance Analysis of key-to-address transformation methods Using an Abstract File Concept. Comm. of ACM 1973, vol. 16, s. 603-612.
- [8] J. Wojdyła: Możliwości i perspektywy automatyzacji projektowania zbiorów i baz danych w ramach laboratorium LABADA. Biuletyn MERA, nr 10/1983, s. 16-27.

mgr inż. TOMASZ ANASZEWICZ  
 dr inż. MIECZYŚLAW OWOC  
 dr JANUSZ WOJDYŁA  
 mgr STANISŁAW ZIMNOCHO  
 Akademia Ekonomiczna—Wrocław

## HDSMAP—PAKIET DO MODELOWANIA ALOKACJI ORAZ PREZENTACJI OPISU BAZ DANYCH SYSTEMU HADES

Proces alokacji bazy danych polega na przydzieleniu /zarezerwowaniu/ obszaru pamięci o bezpośrednim dostępie, odpowiedniego do przechowywania danych użytkownika wraz z danymi sterującymi systemu operacyjnego oraz wszystkich systemów nad nim nadbudowanych, w tym systemu zarządzania bazą danych. Przydzielony obszar pamięci powinien umożliwiać dopisywanie nowych rekordów w miarę ich pojawiania się. Konieczne jest więc uwzględnienie rezerwy pamięci na przechowywanie rekordów pojawiających się w czasie eksploatacji. Jeśli system gospodarki pamięcią umożliwia jej odzyskiwanie po rekordach usuniętych, należy uwzględnić także tę właściwość w procesie alokacji.

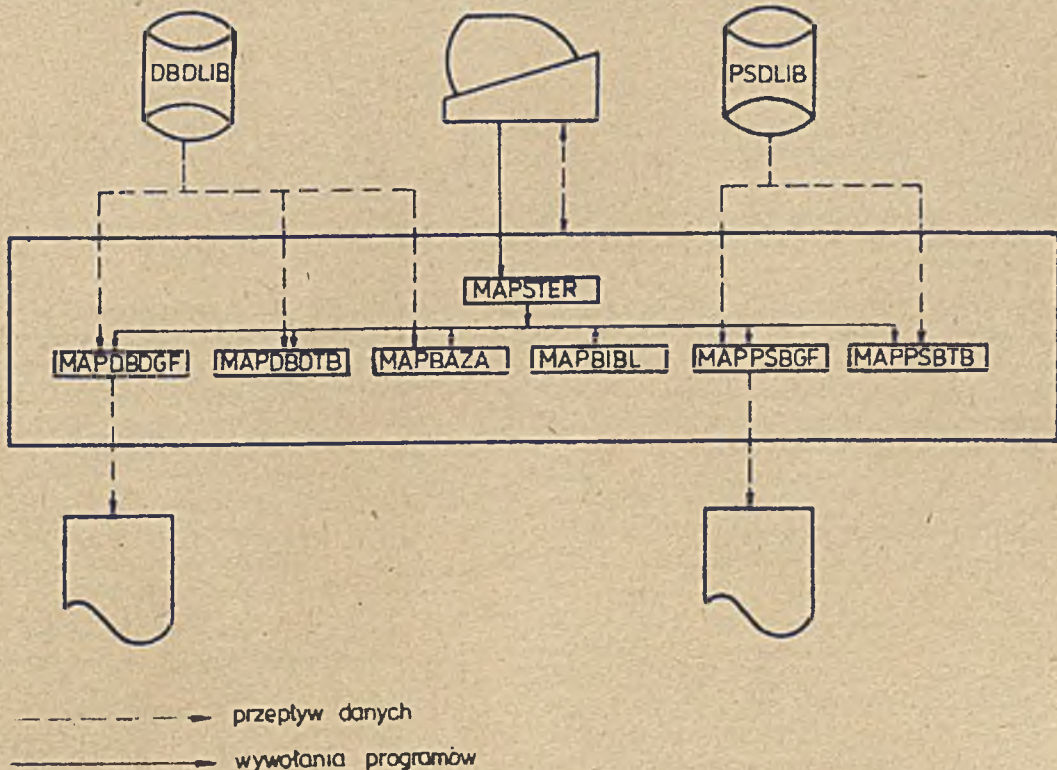
Alokacja bazy danych nie jest realizowana automatycznie. Jest ona jednym z ostatnich etapów procesu projektowania i doskonalenia systemu informatycznego. W tej fazie prac nad tworzeniem systemu informatycznego pojawia się również potrzeba sporządzenia zestawień dokumentujących opis zawartości bazy danych. Wynika stąd połączenie w ramach jednego pakietu dwu funkcji: modelowania procesów alokacji zasobów informacyjnych systemu bazy danych oraz prezentacji graficznej i tabelarycznej opisów bazy danych.

### Ogólna charakterystyka funkcji pakietu

Pakiet HDSMAP jest zestawem siedmiu programów pracujących w trybie konwersacyjnym pod kontrolą systemu operacyjnego OS/JS z opcją TSO /rys. 1./. Oprócz programu MAPSTER, którego zadaniem jest informowanie użytkownika o możliwościach pakietu oraz umożliwienie wyboru żądanej funkcji, każdy z sześciu pozostałych programów realizuje jedną z funkcji pakietu:

MAPBAZA - modelowanie alokacji baz danych typu HSAM, HISAM, HDAM oraz HIDAM wraz z bazą INDEX,  
 MAPCIBL - modelowanie alokacji bibliotek systemu HADES zawierających opisy DBD oraz PSB w postaci źródłowej oraz skompilowanej<sup>1/</sup>  
 MAPDBDGF - prezentacja wybranych opisów DBD w postaci graficznej,  
 MAPPSBGF - prezentacja wybranych opisów PSB w postaci graficznej,  
 MAPDBDTB - prezentacja wybranych opisów DBD w postaci tabelarycznej,  
 MAPPSBTB - prezentacja wybranych opisów PSB w postaci tabelarycznej.

Każdy z programów komunikuje się z operatorem /użytkownikiem/, żądając wprowadzenia określonych danych lub informując o dostępnych możliwościach pracy. Zewnętrznymi źródłami



Rys. 1. Ogólny schemat przetwarzania pakietu HDSMAP

```

.....
:
:                               LISTA KOMEND PROGRAMU                               :
:.....
:LP: NAZWA :   BUDOWA KOMENDY   :   WYNIK DZIAŁANIA KOMENDY   :
: :       :   (SKŁADNIA)       :   (SEMANTYKA)               :
:.....
: 1: HELP  :   HELP/H/?         :   Informacje o budowie i wynikach :
: :       :                   :   działania komend              :
: 2: LIST  :   LIST/L           :   Lista modułów biblioteki DBDLIB :
: 3: FULL  :   FULL/F/**        :   Pełna charakterystyka bazy      :
: 4: SHORT :   SHORT/S/         :   Osobna charakterystyka bazy     :
: 5: DS    :   DS *             :   Pełna charakterystyka zbiorów   :
: :       :   DS IDDD...      :   Pełna char. wybranych zbiorów  :
: :       :                   :   Osobna charakterystyka zbiorów :
: 6: SEG   :   SEG *            :   Pełna charakterystyka segmentów :
: :       :   SEG IDSEG...    :   Pełna char. wybranych segmentów :
: :       :   SEG             :   Osobna charakterystyka segmentów :
:.....

```

```

.....
:
:                               LISTA KOMEND PROGRAMU                               :
:.....
:LP: NAZWA :   BUDOWA KOMENDY   :   WYNIK DZIAŁANIA KOMENDY   :
: :       :   (SKŁADNIA)       :   (SEMANTYKA)               :
:.....
: 7: REL   :   REL * /          :   Pełna char. relacji logicznych  :
: :       :   REL IDSEG...     :   Pełna char. wybranych rel. log. :
: :       :   REL             :   Osobna char. relacji logicznych :
: 8: FLD   :   FLD *           :   Pełna charakterystyka pól       :
: :       :   FLD * IDPOLA... :   Pełna char. wybranych pól       :
: :       :   FLD IDSEG *     :   Pełna char. pól segmentu        :
: :       :   FLD IDSEG IDPOLA... :   Pełna char. wybranych pól seam. :
: :       :   FLD             :   Posortowana lista char. pól     :
: 9: END   :   END/E/+         :   Koniec komend dla aktualnej bazy :
:.....
: Znak / oznacza alternatywy a znaki ... wielokrotne powtórzenie
: Skróty: IDDD, IDSEG, IDPOLA oznaczają odpowiednio identyfikatory
:       zbioru bazy, segmentu i pola
:.....
***

```

Rys. 2. Fragment komend programu MAPDBDTB

danych są ponadto: dla programów MAPBAZA, MAPDBDGF oraz MAPDBDIB - biblioteka skompilowanych DBD oznaczona na rys. 1 nazwą DBDLIB, a dla programów MAPPSBGF oraz MAPPSBTB - biblioteka skompilowanych PSB nazwana PSBLIB.

Programy do prezentacji opisów bazy danych sterowane są prostym językiem komend, za pomocą którego użytkownik definiuje zakres oraz postać danych, które chce uzyskać. W każdym momencie pracy programu istnieje możliwość uzyskania, za pomocą komendy HELP lub znaku zapytania /?/, pełnego wykazu komend dla danej funkcji. Na rys. 2 przedstawiono fragment listy komend programu MAPDBDTB. Rys. 3 przedstawia natomiast wynik działania komen-

dy SEG PRACAR wykonanej przez program MAPDBDTB na DBD opisującym bazę PLACEAR. Analogicznie wyglądają zestawienia generowane przez program MAPPSBTB w odniesieniu do PSB.

Drugi rodzaj zestawień o charakterze dokumentacyjnym stanowią graficzne prezentacje DBD oraz PSB. Graf powiązań segmentów w bazie PLACEAR, wykonany za pomocą programu MAPDBDGF, przedstawiono na rys 4. Symbole używane w tych zestawieniach są wyjaśnione w drukowanych przez programy MAPDBDGF oraz MAPPSBGF legendach. Legendy te wprowadzane są automatycznie po zakończeniu sesji użytkownika z pakietem HDSMAP, o ile korzystano z funkcji graficznych.

Pozostałe dwie funkcje pakietu HDSMAP dotyczą modelowania procesów alokacji. Pierwsza z nich - realizowana przez program MAP.BIBL - dotyczy alokacji bibliotek DBDLIB oraz PSBLIB i powinna być wykorzystywana przez użytkownika podczas instalowania systemu HADES oraz każdorazowo przed rozpoczęciem

<sup>1/</sup> DBD - Data Base Description - opis fizycznej bazy danych w systemie HADES, PSB - Program Specification Block - blok specyfikacji programu - opis fragmentu jednej lub kilku baz fizycznych, na których będzie działał pojedynczy program użytkowy.



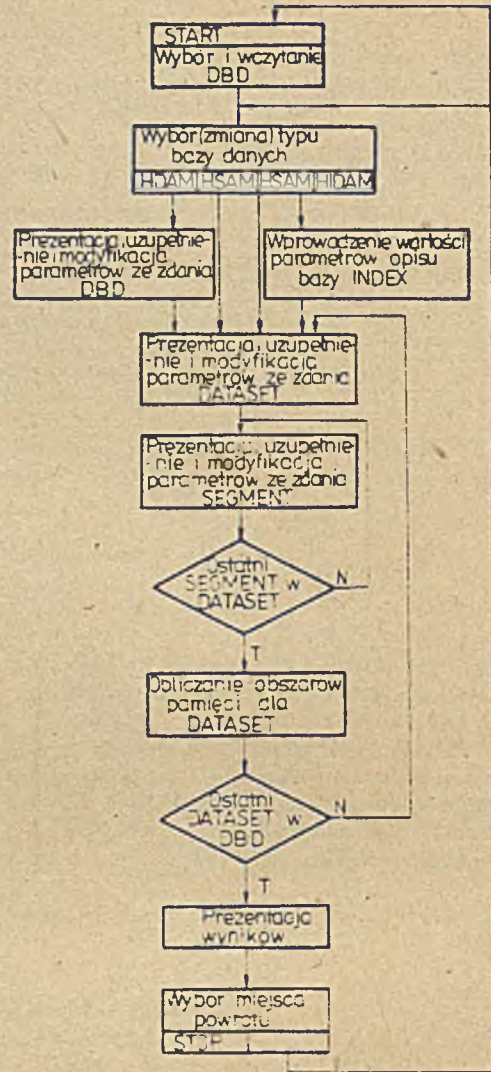


jednostki sterujące, kanały oraz całe instalacje,  
 - łączenie, w ramach określonych jednostek alokacji, danych pełniących różne funkcje w systemie przetwarzania.

Problem właściwej alokacji zbiorów i baz danych został omówiony w dość licznych publikacjach, z których wymienić należy prace J. A. Van der Pool'a dotyczące alokacji zbiorów o organizacji bezpośredniej [8], [9], [10], artykuł H. Mendelzona i inne dotyczące optymalnej alokacji zbiorów sekwencyjnych oraz bibliotecznych [7] czy wreszcie dość liczne prace związane z reorganizacją zbiorów i baz danych, np. [6]. Również dokumentacja systemów operacyjnych i systemów zarządzania bazą danych zawiera opis reguł postępowania oraz wzory dotyczące sporządzania projektu alokacji zbiorów lub baz danych [4], [3], [11], [2], [1].

Na uwagę zasługują także prace związane z komputerowym modelowaniem alokacji zbiorów w systemie operacyjnym OS/JS za pomocą programu ALOKATOR, wchodzącego w skład laboratorium LABADA [12], [14], [13]. Należy jednak dodać, że efektywne stosowanie wszystkich wzorów i algorytmów alokacji jest w praktyce rzadko spotykane, czego wyrazem jest słabe wykorzystanie powierzchni nośników pamięci. Wynika to zarówno z lekceważenia tego problemu przez projektantów, jak i ze złożoności samego procesu alokacji.

Spodziewane efekty automatyzacji prac związanych z alokacją baz danych są następujące:  
 - wybór optymalnych parametrów alokacji z punktu widzenia wykorzystania pamięci zewnętrznej,  
 - możliwość alokacji na podstawie jednostek struktury logicznej danych /segmentów/ lepiej znanych użytkownikowi niż jednostki alokacji /bloki, strony/,  
 - antycypacja zachowania się bazy danych /w



Rys. 6. Ogólny schemat przebiegu modelowania alokacji baz danych za pomocą programu MAPBAZA

\*\* LABADA - HDSMAP. (alokacja bibliotek) \*\*

```
//DBD DD DSN=HDS1.DBDISOUR,UNIT=5061,
// VOL=SER=.....,
// DISP=(,KEEP),DCB=SYS1.MACLIB,
// SPACE=(TRK,(8,,1))
```

a/

\*\* LABADA - HDSMAP (alokacja bibliotek) \*\*

```
//DBDLIB DD DSN=HDS1.DBDLIB,UNIT=5061,
// VOL=SER=.....,
// DISP=(,KEEP),DCB=SYS1.LINKLIB,
// SPACE=(TRK,(4,,1))
```

b/

Rys. 5. Przykładowe wyniki pracy programu MAPBIBL: a - zdanie DD do alokacji biblioteki źródłowych DBD, b - zdanie DD do alokacji biblioteki skomplikowanych DBD

Wykaz parametrów szczegółowych modyfikowanych  
w programie MAPBAZA

Tabela 1

Lp.	Nazwa parametru	Symbol parametru	Typy bazy				
			HSAM	HISAM	HDAM	HIDAM	INDEX /razem z HIDAM/
1.	Zdanie DBD						
	Typy bazy	ACCESS	+	+	+	+	
	Liczba punktów zakotwiczenia	LPUNZAK		+			
	Wielkość obszaru adresowego	MAXBLOCK		+			
	Liczba bajtów w bloku dla segmentów wierzchołkowych	MAXBYTES		+			
2.	Zdanie DATASET						
	Liczba bloków pozostawionych przy ładowaniu	LWOLB			+	+	
	Procent wolnego miejsca pozostawionego przy ładowaniu	WSPWYP			+	+	
	Typ urządzenia pamięci	TURZ	+	+	+	+	+
	Długość bloku zbioru DD1	DBDD1	+	+	+	+	+
	Długość rekordu zbioru DD1	DRDD1		+			
	Długość bloku zbioru OVFLW	DBOVFLW		+			
	Długość rekordu zbioru OVFLW	DROVFLW		+			
3.	Zdanie SEGMENT						
	Liczba wystąpień segmentu w bazie	LWYSSE	+	+	+	+	
	Długość łączników fizycznych	DLAFIZ			+	+	

zakresie parametrów alokacji/ w przypadku zmian wartości charakterystyk ilościowych lub zmian metody organizacji przechowywania danych /np. dodatkowe łączniki, indeksowanie wtórne, łańcuchy randomizacji itp/.

Praca programu MAPBAZA odbywa się w trzech fazach /rys 6/. W pierwszej następuje kompletowanie danych do obliczeń. Podstawowym zasobem danych jest w tej fazie biblioteka DBDLIB, zawierająca DBD w postaci skompilowanej. Po wybraniu przez użytkownika i wczytaniu do programu określonego opisu następuje prezentacja parametrów szczegółowych których wartości mogą być przez użytkownika modyfikowane. Wykaz tych parametrów znajduje się w tabeli 1. Są one pogrupowane wg źródła pochodzenia, a więc według maksymalnej liczby wystąpień. Fragment dialogu z fazy kompletowania parametrów przedstawiono na rys 7.

Druga faza pracy programu MAPBAZA obejmuje obliczenia obszarów pamięci potrzebnych do zaalokowania poszczególnych grup zbiorów danych opisanych za pomocą zdań DATASET. W tym celu wykorzystano między innymi następujące wzory:

$$LBL = \text{FLOOR} \left( \frac{DLREK}{DB-DCOB} \right) \quad /1/$$

gdzie:

LBL - liczba bloków w wybranym obszarze bazy danych /zbiór OSAM lub obszar PRIME - bez indeksu ścieżek - zbioru ISAM/,  
DLREK - długość rekordu bazy danych wyznaczona za pomocą wzoru:

$$DLREK = \sum_{I=1}^{LSEDS} \left( \frac{DSE/I + DPF\%I}{LWYSSE/I + FIT} \right) \quad /2/$$

LSEDS - liczba typów segmentów w analizowanej grupie zbiorów danych,

\* HD5MAF-MAPBAZA \* DD1=PLACEAR ACCESS=HISAM data=27.02.85 czas=18.40.47  
 Numer grupy zbiorow= 1 DD1=PLACEAR OVFLW=DD20V

```

* typ urzadzenia pamieci dyskowej = 5061
* dlugosc bloku dla DD1 = 1690.00
* dlugosc rekordu dla DD1 = 8
777
musi byc dzielnikiem dlugosci bloku
* dlugosc bloku dla DD1 = 1690.00
845i

Zamiast liczby lub znakow: + albo ? wprowadziles znak "I".
Musisz to poprawic i podac :
* dlugosc bloku dla DD1 = 1690.00
845
  
```

Rys. 7. Fragment dialogu programu MAPBAZA z użytkownikiem podczas kompletowania parametrów

DSE/I/ - długość danych I-tego segmentu w analizowanej grupie zbiorów danych,  
 DPFX/I/ - długość prefiksu I-tego segmentu w analizowanej grupie zbiorów danych,  
 L.WYSSE/I/ - liczba wystąpień I-tego segmentu w analizowanej grupie zbiorów danych,  
 DB - długość bloku,  
 DCOB - długość części organizacyjnej bloku /wskaźniki/,  
 FIT - przewidywana długość nie wykorzystanej części bloków, wynikająca z niepodzielności segmentów oraz końcówek rekordu w ostatnim bloku,

W bazach typu HD liczba bloków jest powiększona o narzuty wynikające z parametrów:  
 MAXBLOCK - wielkość obszaru adresowego bazy HDAM,  
 MAXBYTES - liczba bajtów rekordu w blokach obszaru adresowego bazy HDAM,  
 LWOLB - liczba wolnych bloków /HDAM, HIDAM/  
 WSPWYP - współczynnik wypełnienia bloków podczas ładowania bazy /HDAM, HIDAM/.

Do obliczenia obszaru pamięci dyskowej wyrażonej w ścieżkach użyto następujących wzorów:

$$LSC = CEIL (LBL/LBLSC) \quad /3/$$

gdzie:

LSC - liczba ścieżek zajmowanych przez analizowany obszar /np. obszar PRIME w ISAM, obszar OSAM, obszar INDEX w ISAM/,

LBL - liczba bloków w analizowanym obszarze /wyliczona ze wzoru /1/

$$LBLSC = 1 + FLOOR ((DSC - BN) / BI) \quad /4/$$

$$BN = DDA + DKL + XC * SIGN (DKL) \quad /5/$$

$$BI = XA + CEIL (XB * DDA + DKL) + XC = SIGN (DKL) \quad /6/$$

DDA - długość danych w bloku,  
 DKL - długość klucza w bloku /gdy bloki bez klucza przyjmujemy DKL=0/,

SIGN - funkcja określona na liczbach naturalnych następująco:

$$SIGN/X/ = \begin{cases} 0 & \text{gdy } X = \emptyset \\ 1 & \text{gdy } X \neq \emptyset \end{cases}$$

natomiast pozostałe parametry poszczególnych typów pamięci dyskowej podano w tabeli 2.

Narzuty pamięci dla poszczególnych grup zbiorów danych są obliczane następująco:  
 1/ narzut systemowy:

Wykaz parametrów pamięci dyskowej do obliczania obszaru zbioru danych

Tabela 2

Typ pamięci	Długość ścieżki DSC	Parametry			XA	XB	XC
		Liczba cylindrów w woluminie LCYLWOL	Liczba ścieżek w cylindrze LSCCYL				
EC 5052	3625	200	10	61	$\frac{537}{512}$	20	
EC 5061	7294	200	20	101	$\frac{534}{512}$	45	
EC 5066	13165	404	19	135	1	56	

\* HDSMAP-MAPBAZA \* DBD=PLACEAR ACCESS=HIDAM data=27.02.85 czas=18.31.19  
wyniki modelowania

```

.....
: Nr      : Nazwa  : Typ  : Dlug. : Obszar : Narzut w % : Wspolcz. : Wsp. :
: grupy  : DD     : pam. : bloku : zbioru : ..... : pustych  : woln. :
: zbior. :        : zewn. :      : (TRK)  : syst. : tech. : blokow  : msc. :
: 1 : PLACEAR : 5061 : 1690 : 2952 : 39.7 : 7.3 : 5 : 20 :
.....

```

a/

\* HDSMAP-MAPBAZA \* DBD=PLACEAR ACCESS=HIDAM data=27.02.85 czas=18.37.35  
wyniki modelowania bazy INDEX=INDEXS

```

//DD11 DD DSN=.....(PRIME),DISP=(,KEEP),
// UNIT=5061,VOL=SER=.....,SPACE=(CYL,(6)),
// DCB=DSORG=IS

```

b/

\* HDSMAP-MAPBAZA \* DBD=PLACEAR ACCESS=HISAM data=27.02.85 czas=18.44.30  
wyniki modelowania

```

.....
Dane wejsciowe : Wyniki modelowania
.....
Numer grupy zbiorow= 1 : NB TRK CYL
:
Typ pamieci zewn. = 5061 : obszar ISAM(INDEX) 7.12 -
: Obszar ISAM(PRIME) 3013.05 -
Obszar glowny(ISAM)= PLACEAR : Obszar USAM - 2161 -
- dlugosc bloku = 1690 :
- dlugosc rekordu = 845 : Razem ISAM - - 23
: Razem ISAM+USAM - 2621 131.0
Obszar nadm.(OSAM) = DD20V :
- dlugosc bloku = 1690 : Narzut systemowy = 26.4 %
- dlugosc rekordu = 845 : Narzut technologiczny = 14.5 %
.....

```

c/

Rys. 8. Wyniki pracy programu MAPBAZA: a - tabela dla obszarów OSAM baz HD i HSAM, b - zdanie DD dla bazy INDEX, c - tabela dla baz HISAM

$$\text{NARZSYST} = \frac{\text{DS}-\text{DI}}{\text{DS}} \times 100\% \quad /7/$$

gdzie:  
DI - dane informacyjne  
DS - dane systemowe

2/ narzut technologiczny

$$\text{NARZTECHN} = \frac{\text{DT}-\text{DS}}{\text{DT}} \times 100\% \quad /8/$$

gdzie:  
DT - dane technologiczne.

Poszczególne rodzaje danych liczone są z wzorów:

$$\text{DI} = \sum_{I=1}^{\text{LSEDS}} \text{DSE}(I) \times \text{LWYSSE}(I) \quad /9/$$

$$\text{DS} = \sum_{I=1}^{\text{LOBDS}} \text{LBL}(I) \times \text{DB}(I) \quad /10/$$

$$\text{DT} = \sum_{I=1}^{\text{LOBDS}} \text{LSC}(I) \times \text{DSC} \quad /11/$$

gdzie:  
LOBBAZ - liczba obszarów w analizowanej

grupie zbiorów danych /PRIME, INDEX bazy HISAM i bazy INDEX, obszar OSAM/.

Wyliczenie narzutu systemowego oraz technologicznego pozwala ocenić projekt struktury fizycznej bazy danych z punktu widzenia wykorzystania pamięci dyskowej. Narzut systemowy informuje o tym, jaka część obszaru zajętego przez bloki wykorzystywana jest na dane systemu zarządzania bazą danych /łączniki adresowe, liczniki, mapy bitowe itp./ lub jest całkowicie niewykorzystana np. z powodu niedostosowania długości segmentów do długości bloków. Narzut technologiczny natomiast obliczony jest w celu umożliwienia oceny projektu pod kątem strat wynikłych z powodu niedostosowania długości bloku do długości ścieżki lub ograniczeń narzuconych przez metodę dostępu ISAM /przydział obszarów w całym cylindrach/.

Ostatnią fazą pracy programu MAPBAZA jest wyprowadzenie wyników obliczeń. Wyniki są wyprowadzane na ekran końcówki w dwóch postaciach: tabelarycznej oraz zdań DD /rys.8/.

Dzięki temu użytkownik pakietu ma możliwość włączenia do dokumentacji projektowej systemu wyników obliczeń w postaci zbliżonej do standardu dokumentacji i jednocześnie przekazać do eksploatacji gotowe zdania DD - po wypełnieniu na ekranie brakujących nazw oraz wydrukowaniu zawartości ekranu na urządzeniu trwałej kopii.

#### Zastosowanie pakietu HDSMAP w procesie projektowania hierarchicznych baz danych

Pakiet HDSMAP winien być stosowany w końcowym etapie prac drugiej fazy tworzenia systemu bazy danych, a mianowicie w projektowaniu struktury konstrukcyjno-technologicznej. Ułatwia on wówczas obliczenia parametrów alokacyjnych dla nowo tworzonych baz danych w standardzie SZBD HADES. Obliczenie informacyjnego, systemowego oraz technologicznego obszaru bazy danych może być również pomocne w pracy administratora szczególnie wówczas, gdy zachodzi potrzeba reorganizacji bazy lub zmiany urządzeń pamięci o dostępie bezpośrednim. Znając dynamikę liczebności segmentów w bazie można za pomocą pakietu HDSMAP oszacować maksymalną pojemność uprzednio zaalokowanych obszarów bazy lub określić przewidywany termin jej reorganizacji. Zastosowanie pakietu do obliczenia obszarów bibliotek DBDLIB oraz PSBLIB pozwala na zaoszczędzenie powierzchni 2-3 cylindrów urządzenia EC-5061.

Drugą ważną grupą funkcji pakietu HDSMAP jest możliwość graficznej i tabelarycznej prezentacji opisów DBD oraz PSB. Specyfika opisu DBD lub PSB sprawia, że są one zrozumiałe jedynie dla informatyków - specjalistów. Nie mogą zatem stanowić materiału informacyjnego lub dokumentacyjnego we współpracy ze specjalistami "branżowymi". Wydaje się, że graficzna forma opisu daje ogólną orientację o strukturze i liczebności bazy, natomiast forma tabelaryczna pozwala na zapoznanie się z charakterystyką dowolnych jej elementów, wyrażoną w komunikatywnym języku naturalnym. Ze względu na potrzebę posiadania odpowiedniego opisu DBD lub PSB opcje opisu graficznego lub tabelarycznego mogą być używane już na etapie generowania pierwszych wersji struktur bazy HADES, otrzymywanych za pomocą pakietu HDSLISA, w końcowym etapie prac fazy modelowania conceptualnego dziedziny przedmiotowej.

#### L i t e r a t u r a :

- [1] HADES - System Zarządzania Hierarchiczną Bazą Danych. Opis Zastosowań. IKSAiP, Wrocław 1983.
- [2] IDMS Part 5: Database Design. ICL, 1982.
- [3] Information Management System/360, Version 2, System/Application Design Guide, IBM Corp., 1974.
- [4] Introduction to IBM System/360 Direct Access Storage Devices and Organization Methods. Form nr GD 20-1649-4, IBM Corp., 1973.
- [5] HDSMAP - Pakiet do modelowania procesów alokacji oraz prezentacji opisu hierarchicznej bazy danych. Raport badawczy Instytutu Informatyki nr II-5-84, AE Wrocław, 1984.
- [6] K. Maruyama, S. E. Smith: Optimal Reorganization of Distributed Space Disk Files, Communications of the ACM, nr 11/1976, pp. 634-642.
- [7] H. Mendelson, J. S. Pliskin, U. Yechiali: Optimal Storage Allocation for Serial Files. Communications of the ACM, nr 1/1979, pp. 124-130.
- [8] J. A. Van der Pool: Optimum Storage Allocation for Initial Loading of a File, IBM Journal of Research and Development, nr 6/1972, pp. 579-584.
- [9] J. A. Van der Pool: Optimum Storage Allocation for a File in Steady State, IBM Journal of Research and Development, nr 1/1973, pp. 27-38.
- [10] J. A. Van der Pool: Optimum Storage Allocation for a File with Open Addressing. IBM Journal of Research and Development, nr 3/1973, pp. 106-114.
- [11] Uniwersalny system zarządzania bazą danych RODAN. Dokumentacja użytkowa, część I, CPiZI, Warszawa 1979.
- [12] J. Wojdyła: Możliwości i perspektywy automatyzacji projektowania zbiorów i baz danych w ramach laboratorium LABADA, Biuletyn Techniczno-Informacyjny MERA, nr 11/1983.
- [13] J. Wojdyła, S. Zimnocho: Construction problems of Computer Aided Data Base Allocators. Proceedings of VII-th International Seminar on Data Base Management Systems, Varna 1984, s. 185-192.
- [14] S. Zimnocho: Wspomagane komputerem projektowanie alokacji dyskowych zbiorów danych. Materiały V Międzynarodowej Konferencji Naukowo-Technicznej nt. "Komputerowe wspomaganie projektowania", TNOiK, Poznań 1983.

## SYSTEMY SKOT/VS I HADES/VS - BAZOWE OPROGRAMOWANIE NARZĘDZIOWE DLA M.C. JS DRUGIEGO POKOLENIA

Spodziewane wprowadzenie do produkcji przez ZE ELWRO maszyny cyfrowej Jednolitego Systemu drugiego pokolenia /wyposażonej w pamięć wirtualną/, nazwanej R-34, spowodowało podjęcie w IKSAiP prac nad przygotowaniem oprogramowania narzędziowego dla tej maszyny. Ustalono, że w pierwszym okresie opierać się ono będzie na systemach:

- SKOT/VS - system transakcyjny umożliwiający budowę i eksploatację wielodostępnych systemów informatycznych, opartych o układy teleprzetwarzania,
- HADES/VS - System Zarządzania Hierarchiczną Bazą Danych,
- współpracujących ze SKOT/VS i HADES/VS, wspomagających procesy tworzenia i eksploatacji baz danych, wyszukiwania informacji baz danych, przygotowywania oprogramowania użytkowego itp.

Systemy te będą dostępne już od momentu rozpoczęcia sprzedaży R-34. Systemy SKOT/VS i HADES/VS stanowiąc będą funkcjonalną ciągłość systemów SKOT i HADES, rozprowadzanych już przez ELWRO-Serwis, pracujących na R-32.

### System Kontroli i Obsługi Terminali SKOT/VS

System SKOT/VS stanowi funkcjonalne rozwinięcie Systemu SKOT pracującego na R-32 pod kontrolą Systemu Operacyjnego OS/JS. Niniejszy artykuł sygnalizuje udogodnienia zawarte w systemie SKOT/VS w stosunku do systemu SKOT. SKOT/VS może pracować pod kontrolą systemów operacyjnych VS/JS, OS/VS1 lub OS/VS2. Używaną metodą dostępu do zbioru jest VSAM. Metodami telekomunikacyjnymi są BTAM lub TCAM /podobnie jak w SKOT/ oraz wirtualna metoda telekomunikacyjna VTAM. Metoda VTAM powoduje przeniesienie poważnej części operacji sterowania siecią terminali do programu umieszczonego w procesorze telekomunikacyjnym, odciążając w ten sposób komputer centralny. Ważną cechą VTAM jest użycie protokołu SDLC do transmisji danych magistralami, łączącymi procesory telekomunikacyjne i niektóre nowe typy terminali.

SKOT/VS zapewnia wiele udogodnień pozwalających na dużą efektywność systemów użytkowych. Przykładem jest prowadzenie kroniki pracy systemu /nowość w stosunku do systemu SKOT/. SKOT/VS może być tak wygenerowany, aby zawierał procedury umożliwiające podjęcie próby odtworzenia pamięci w przypadku uszkodzenia systemu. W przypadku nienor-

malnego zakończenia pracy systemu istnieje możliwość odtworzenia go do wymaganej postaci. Ponowne uruchomienie systemu SKOT/VS zależy od stanu, w jakim zakończyła się poprzednia sesja.

- Istnieją trzy typy startu systemu SKOT/VS:
- a/ start niezależny od poprzedniego zakończenia sesji,
  - b/ start będący kontynuacją poprzedniej sesji, tzn. system SKOT/VS jest przed startem doprowadzony do stanu, w jakim znajdował się w trakcie poprzedniego kontrolowanego zamknięcia,
  - c/ start systemu po upadku systemu - w tym przypadku system SKOT/VS doprowadzony jest do stanu, w jakim znajdował się w określonych punktach przed niekontrolowanym wyłączeniem /takie punkty określa programista systemowy w trakcie przygotowywania systemu SKOT/VS do konkretnego zastosowania/.

System SKOT/VS może obsługiwać sieć zawierającą różne typy terminali. Jednym z terminali może być komputer, na którym zainstalowany jest inny SKOT/VS. Wykorzystując łączność międzysystemową istnieje możliwość wykorzystania zasobów jednego systemu użytkowego przez inny system użytkowy. Rozproszenie zasobów systemowych umożliwia ich wykorzystanie bez konieczności żądania ich wyłączności dla danego komputera. W ten sposób można wykorzystywać np. terminale, zbiory baz danych, programy użytkowe. W systemie SKOT/VS zapewniony jest sterowany dostęp do własnej bazy danych bardzo prostej oraz przez specjalne łącza programowe do bazy danych zarządzanej przez system HADES/VS.

### System HADES/VS

System Zarządzania Hierarchiczną Bazą Danych HADES/VS jest gotowym oprogramowaniem umożliwiającym tworzenie i użytkowanie baz danych w informatycznych systemach zarządzania. System HADES/VS jest rozwinięciem systemu HADES i umożliwia pracę w środowisku wirtualnym na komputerach JSEMC drugiego pokolenia /np. R-55 i R-34, R-60/. HADES/VS zezwala na realizację prac w trybie wsadowym, a poprzez łącze programowe z Systemem Kontroli i Obsługi Terminali SKOT/VS w trybie bezpośrednim. System dostarcza swoim użytkownikom wiele narzędzi programowych ułatwiających zapisywanie, aktualizację i wyszukiwanie danych w dużych scentralizowanych bazach. Bazy te tworzone są z myślą o stale rosnących potrze-

bach informatycznych np. z zakresu zarządzania produkcją, zapasami, gromadzenia i przetwarzania danych o pacjentach, studentach itp., rezerwacji miejsc w hotelach, samolotach, autobusach i pociągach.

Cechami charakterystycznymi systemu HADES/VS są:

- otwartość strukturalna systemu umożliwiająca dalszą jego rozbudowę,
- dane przechowywane w bazie mogą być stałej lub zmiennej długości,
- powiązania między danymi w kilku bazach danych umożliwiające nieredundantne ich przechowywanie,
- niezależność programów od fizycznej organizacji danych,
- restrukturyzacja i reorganizacja istniejących baz danych nie wymagająca modyfikacji programów użytkowych,
- możliwość odtwarzania bazy danych,
- możliwość prowadzenia kontroli systemu,
- programy użytkowe mogą być pisane w językach PL/I, COBOL lub ASSEMBLER, w których można odwołać się do subjęzyka danych DL/I.

Język DL/I jest integralną częścią systemu HADES/VS, pełni on funkcję języka manipulowania danymi. Program użytkowy posiada dwa rodzaje powiązań z językiem DL/I; pierwsze poprzez opis bazy / tzn. opis logicznej struktury bazy danych - nie stanowiący integralnej części programu/, drugie poprzez symboliczne wskazanie na żądane operacje we-wy, które ma zrealizować język DL/I w czasie pracy programu użytkowego. Podczas przetwarzania bazy danych dla określonego zadania ma się do czynienia jedynie z tymi danymi, które zostały wcześniej zdefiniowane jako istotne, związane z określonym zastosowaniem. Każdy program użytkowy korzystający z bazy danych może mieć własny unikalny podzbiór danych.

Język DL/I obsługuje dwie podstawowe organizacje pamięci: hierarchiczną sekwencyjną /HS/ oraz hierarchiczną bezpośrednią /HD/.

W ramach organizacji hierarchicznej sekwencyjnej działają następujące metody dostępu:

- HSAM hierarchiczna sekwencyjna metoda dostępu,
- HISAM hierarchiczna indeksowo-sekwencyjna metoda dostępu,
- SHSAM uproszczona hierarchiczna sekwencyjna metoda dostępu,

- SHISAM uproszczona hierarchiczna indeksowo-sekwencyjna metoda dostępu.

Dla organizacji hierarchicznej bezpośredniej dostępne są następujące metody dostępu do zbiorów:

- HDAM hierarchiczna bezpośrednia metoda dostępu,
- HIDAM hierarchiczna indeksowo-bezpośrednia metoda dostępu.

Najistotniejszymi różnicami między organizacją sekwencyjną a bezpośrednią jest sposób wiązania segmentów oraz techniki wyszukiwania. Oprócz wymienionych dwóch organizacji pamięci, DL/I obsługuje też zbiory niehierarchiczne - do tego celu służy metoda dostępu GSAM /uogólniona sekwencyjna metoda dostępu/.

System HADES/VS, podobnie jak jego poprzednia wersja, pracująca na maszynach Jednolitego Systemu pierwszej generacji, wyposażony jest w wiele programów pomocniczych realizujących m. in. takie funkcje jak:

- generowanie systemu,
- generowanie opisu bazy danych,
- generowanie bloków specyfikacji programów,
- utrzymywanie bloków sterujących zastosowaniami,
- reorganizację bazy danych,
- kopiowanie i składowanie bazy danych,
- odtwarzanie bazy danych.

Różnice między systemem HADES a HADES/VS polegają m. in. na tym, że ten ostatni system wyposażony jest dodatkowo w:

- metody dostępu GSAM, SHSAM, SHISAM,
- możliwość wtórnej indeksacji,
- możliwość obsługi zbiorów VSAM, zbiorów ze zmienną długością rekordów oraz możliwość kompresji zbiorów.

Użytkownik eksploatujący bazę danych założoną w oparciu o system HADES ma zapewnić proste przejście na eksploatację swojej bazy pod systemem HADES/VS.

Wymagania sprzętowo-programowe systemu HADES/VS:

- system operacyjny VS/JS bądź OS/VS1, OS/VS2 z SVC typu 4,
- region pamięci nie mniejszy niż 2 MB,
- dwa pakiety dyskowe po 30 Mb lub jeden 100 Mb /bądź większy/ dla systemu i bazy danych.

Ilość pakietów dyskowych zależy od wielkości bazy, w artykule podano ilości minimalne.



W dniach 28, 29 maja 1985 r. w Poznaniu odbyła się konferencja pod powyższym tytułem zorganizowana przez Towarzystwo Naukowe Organizacji i Kierownictwa - Wielkopolski Oddział w Poznaniu i Akademię Ekonomiczną w Poznaniu.

W pracach konferencji opublikowano jedenaście referatów, fragmenty których przedstawiono poniżej.

1. prof. dr. hab. Henryk Sobis, Akademia Ekonomiczna we Wrocławiu: "Organizacja systemów informatycznych dla małych i średnich przedsiębiorstw".

... "W kraju obecnie produkowanych jest kilka typów mikrokomputerów: systemy MERA 100, MERA 200 wytwarzane w Zakładach MERA-BŁONIE oraz PSPD-90 produkcji MERA-KFAP w Krakowie i mikrokomputery serii ELWRO 500 i ELWRO 600 produkowane w ZE ELWRO we Wrocławiu. Ponadto spośród najnowszych modeli tego sprzętu na uwagę zasługują także: mikrokomputer MERITUM I produkcji Zakładów MERA-ELZAB oraz mikrokomputer MK45 produkowany przez MERA-KFAP.

Na najbliższe lata poważne miejsce na naszym rynku mikroprocesorowym powinien zająć system ELWRO 500, w tym ELWRO 513 i ELWRO 523. Rodzina tych mikroprocesorów weszła na nasz rynek w końcu roku 1984, jednak jeszcze w bardzo skromnych ilościach. Obecnie ZE ELWRO przygotowują wzbogaconą wersję systemu mikrokomputerowego o nazwie ELWRO 600, która w końcu tego roku powinna ukazać się w wersji informacyjnej na naszym rynku. Sądzić można, że wspomniany sprzęt produkcji ZE ELWRO jest w stanie w najbliższych latach zaspokoić potrzeby naszego rynku.

2. doc. dr Stefan Abt, Instytut Gospodarki Magazynowej w Poznaniu: "Budowa obiektowych systemów informatycznych a makrosystemy".

... "W latach 70-tych w Polsce, kiedy dominował scentralizowany system zarządzania gospodarką narodową zrodziła się koncepcja Krajowego Systemu Informatycznego /KSI/, pochłaniająca znaczne nakłady finansowe oraz nakłady sił i środków, zmierzająca do budowy tzw. infostrady i podłączonych do niej różnego typu systemów, a w szczególności systemów, które dziś można nazwać makrosystemami takimi jak: SPIS, SEIF, PESEL, TEREN, ŚWIATOWID, WEKTOR, MAGMA, MERKURY, HERKULES, TRAKT, SOKRATES, CENPLAN, RESPLAN, REGPLAN. Przewidywano nawet podłączenie się do nich obywateli. Wizja ta okazała się w praktyce nierealną, a dziś wobec reformy gospodarczej lat 80-tych w Polsce, mającej w swoich założeniach decentralizację zarządzania gospodarką - większość tych prac została wstrzymana. Natomiast te, które były zaawansowane we

wdrożeniach, wymagają gruntownej przebudowy, zwłaszcza w zakresie metod zarządzania, jak również przetwarzanych informacji. Załamaniem się gospodarki sprawiło nadto, że dotychczasowe prognozy rozwoju informatyki okazały się całkowicie nierealne i obecnie skromne możliwości w tym zakresie wymagają nowego spojrzenia na rozwój informatyki w Polsce. Z owych wielkich systemów pozostały w zasadzie: SPIS /System Państwowej Informacji Statystycznej/, gruntownie zmieniany CENPLAN /obejmujący reformowane również planowanie gospodarki narodowej/ oraz częściowo wykorzystywany system PESEL /Powszechny Elektroniczny System Ewidencji Ludności/".

3. mgr Grzegorz Gruchman, ZETO - Poznań: "Teoria i praktyka zastosowań technologii bazy danych".

4. mgr Stefan Kuczyński, WIEPOFAMA - Poznań: "Bariery organizacyjne tworzenia baz i banków danych".

... "Tworzenie baz i banków danych jest procesem długotrwałym i uzależnionym od wielu czynników. Większość trudności, które można napotkać podczas tej pracy jest przeważnie niezależna od osób wykonujących ją. Najważniejszymi barierami organizacyjnymi są:

- nieodpowiedni sprzęt komputerowy,
- kwalifikacje personelu,
- niedostateczne przygotowanie przedsiębiorstwa i konieczność wykorzystywania kodów i indeksów zewnętrznych,
- brak metodyki projektowania ukierunkowanej na SZBD,
- błędy projektowe i niedopracowanie modelu pojęciowego,
- zła organizacja prac projektowych,
- bariera psychologiczna".

5. dr inż. Jerzy Dyczkowski, Zrzeszenia MERA, Warszawa: "Sprzęt komputerowy w Polsce. Dostawy a zapotrzebowanie".

6. dr inż. Grzegorz Bartoszewicz, Akademia Ekonomiczna w Poznaniu: "Kierunki rozwoju architektury systemów komputerowych".

... "Procesor czołowy spełnia następujące funkcje:

- /1/. przejmuje zadania procesora centralnego związane z transmisją danych,
- /2/. pozwala na dynamiczne analizowanie, definiowanie i modyfikowanie sieci,
- /3/. staje się buforem zapewniającym efektywną /szybką/ współpracę z użytkownikami terminalnymi.

Z kolei do zadań procesora zadaniowego /tylnego/ należą:

- /1/. przetwarzanie transakcji w systemach zarządzania zbiorami, zarządzanie bazą danych,
- /2/. zapewnienie bezpośredniego wejścia do systemu, zarówno w ramach sieci lokalnej jak i szerszej sieci terytorialnej,

/3/ czasowe przejęcie funkcji procesora głównego w przypadkach awarii".

7. doc. dr hab. inż. Zbigniew Kierzkowski, Politechnika Poznańska: "Zasady budowy systemów informatycznych dla określonych potrzeb".

... "Rozwiązywanie nowych problemów zmierzającego do uzyskiwania oprogramowania systemów informatycznych o takich własnościach, że:

1. Zasoby użytkowe wykorzystywane we wspomagananiu - programy i dane - odwzorowują komputerowo dziedzinę przedmiotową /wiedzę/

2. Systemy są otwarte, tzn. ich zasoby użytkowe mogą być rozszerzane. Możliwe jest tworzenie /i rozbudowa/ systemu, ponieważ zasoby traktuje się jako inwarianty, a modele operowania na tych inwariantach /i odpowiadające oprogramowanie inwariantowe/ ułatwiają budowę oprogramowania systemu.

3. Język dialogu użytkownika z komputerem zapewnia opis zasobów i opis procesu wspomaganiania. Proponowane rozwiązania można oceniać analizując własności trybu dialogowego względem przesłanej /możliwych do uzasadnienia przez porównanie znanych rozwiązań z rozwiązaniami proponowanymi/.

8. dr Piotr Refermat, Akademia Ekonomiczna w Poznaniu: "Struktura zawodów informatycznych a struktura kształcenia informatyków w Polsce".

... "Zawody związane z eksploatacją komputera podlegają w skali mikro redukcji ilościowej. Spowodowane to jest następującymi przyczynami:

- zmniejsza się udział przetwarzania wsadowego, które wymagało licznych manipulacji taśmami magnetycznymi, drukarkami, czytnikami kart i taśm na korzyść przetwarzania konwersacyjnego, które wymaga pamięci dyskowych podłączonych na stałe do systemu komputerowego.

- zmniejszenie zmienności wykorzystania sprzętu w wyniku jego potaniaenia zmniejsza zapotrzebowanie na operatorów,  
- nowe generacje drukarek /drukarki laserowe/, bardzo pojemne pamięci dyskowe oraz coraz doskonalsze systemy operacyjne i ich hardware'owa realizacja w postaci mikroprocesorów zmniejsza również liczbę operatorów obsługujących daną konfigurację komputera".

9. dr inż. Tomasz Pawlak, Sekretariat Komitetu Informatyki: "Rola i zadania centrum w sterowaniu rozwojem informatyki".

... "Nowa sytuacja prawna w zakresie centralnego sterowania rozwojem informatyki powstała po przyjęciu przez Sejm PRL ustawy z dnia 3 grudnia 1984 r. o utworzeniu Komitetu do Spraw Nauki i Postępu Technicznego przy Radzie Ministrów oraz Urzędu Postępu Naukowo-Technicznego i Wdrożeń. Organy te, obok sterowania badaniami naukowymi i pracami rozwojowymi oraz wdrożeniami ich wyników do praktyki gospodarczej - mają również programować i koordynować rozwój i wdrażanie informatyki do gospodarki narodowej. Można spo-

dziewać się, że jednocześnie z wydaniem w kwietniu 1985 r. Rozporządzenia Rady Ministrów w sprawie trybu i terminów przekazania spraw należących dotychczas do właściwości Ministra Nauki, Szkolnictwa Wyszczególnienia i Techniki - do prowadzenia przez Komitet do Spraw Nauki i Postępu Technicznego przy RM i Urząd Postępu Naukowo-Technicznego i Wdrożeń - zostanie zniesiony Komitet Informatyki, a jego Sekretariat - włączony do Urzędu Postępu Naukowo-Technicznego i Wdrożeń. Rozpoczęcie działalności w nowym układzie organizacyjnym można przewidywać od początku II półrocza 1985 r.

10. mgr inż. Jerzy Frączak, mgr inż. Roman Janeki, Centrum Badawczo-Wdrożeniowe MERCOMP - Poznań: "Możliwości wykorzystania lokalnych sieci komputerowych w przetwarzaniu danych".

... "Opisane wyżej właściwości sieci lokalnych, a szczególnie możliwości modułowej ich konstrukcji mogą być wykorzystane przy organizacji przetwarzania danych w sposób różniący się od przetwarzania scentralizowanego. W referacie pragniemy zwrócić uwagę na dwa zasadnicze aspekty wykorzystywania sieci lokalnych. Jednym z nich jest możliwość przechowywania i zarządzania zbiorami danych.

Zbiory danych mogą być gromadzone w centralnej, z punktu widzenia sieci lokalnej, bazie danych /baza obsługiwana przez File Server/ lub też rozproszonych u użytkowników /pamięci komputerów osobistych/. Zarządzanie zbiorami danych skupione jest w specjalizowanym urządzeniu końcowym /File Server/.

11. mgr Marek Majewski, ZETO-Poznań: "Potrzeby informacyjne i systemy informatyczne w organizacjach gospodarczych".

... "Potrzeby informacyjne wynikające z przytoczonych wyżej działań przykładowo można ująć w następujące zakresy dotyczące:

- danych o rynku krajowym i zagranicznym /chłonność rynku, wahania cen zarówno wyrobów jak i materiałów, kooperancji - umowy, powiązania kooperacyjne pod kątem cen i kosztów transportu/,

- wariantowania planów produkcji, /zmiany asortymentowe produkcji, bilansowanie potrzeb materiałowych w układzie wariantów planu, dostępność materiałów i surowców, zastępowalność materiałów, bilansowanie nakładów pracy żywej w układzie wariantów planu, ocena wykorzystania zdolności produkcyjnych - efektywność wykorzystania majątku produkcyjnego, analiza rentowności produkcji w układzie wariantów planu/,

- bieżącej kontroli realizacji zadań produkcyjnych,

- optymalizacji gospodarki zasobami produkcji /analiza poziomu zapasów w przekrojach asortymentowych, minimalizacja odpadów i badanie możliwości zagospodarowania powstałych odpadów, racjonalizacja gospodarki remontowej i częściami zamiennymi, urealnianie norm pracy/,

- danych dla celów sprawozdawczości GUS oraz organów finansowych, kontrolnych itp."



CENTRUM NAUKOWO-PRODUKCYJNE  
SYSTEMÓW STEROWANIA

40-153 KATOWICE, ul. Armii Czerwonej 160

## **MERASTER -EKSPORTER SYSTEMÓW MIKROKOMPUTEROWYCH**

← EKSPORT SYSTEMÓW DO  
KRAJÓW SOCJALISTYCZNYCH  
W LATACH 1981-1984



### **ZASTOSOWANIA SYSTEMÓW**

- STANOWISKA AUTOMATYZACJI BADAŃ I EKSPERYMENTÓW NAUKOWYCH
- WIELOFUNKCYJNE STANOWISKA DLA OBSŁUGI PROCESU DYDAKTYCZNEGO
- STANOWISKA ZBIERANIA I PRZEKAZYWANIA DANYCH DLA PRACY AUTOMATYCZNEJ I W SYSTEMACH TELEPRZETWARZANIA
- SIECI KOMPUTEROWE

### **MERASTER OFERUJE:**

- UŻYTKOWE SYSTEMY OPROGRAMOWANIA
- OPROGRAMOWANIE SYSTEMOWE
- USŁUGI SOFTWARE'OWE
- SERWIS

TEL. 597-206 597-086  
TELEKS 031 5958  
mest pl

