

OŚRODEK BADAWCZO-ROZWOJOWY INFORMATYKI

WSPÓLNA BAZA DANYCH III

Europejski
Program Badawczy Diebolda

Zeszyt 32

Warszawa 1972

WSPÓLNA BAZA DANYCH III

Europejski
Program Badawczy Diebolda

WYŁĄCZNIE DO UŻYTKU
NA TERENIE PRL

Zeszyt 32 (E 62)

Tytuł oryginału: COMMON DATA BASE III
Document No E 62

Tłumaczenie: Mgr inż. Franciszek Haratym

Komitet Redakcyjny

Inż. Mieczysław Guld, Mgr inż. Franciszek Haratym, Janina Jerzykowska (Sekretarz),
Dr Stanisław Nelken, (Z-ca Przewodniczącego), Dr Zdzisław Zapolski (Przewodniczący)

Wydawca: OŚRODEK BADAWCZO-ROZWOJOWY INFORMATYKI
Działowy Ośrodek Informacji
Warszawa, ul. Marszałkowska 104 122

S P I S T R E Ś C I

Streszczenie	str.	1
I. Wprowadzenie	"	3
A. Wspólna baza danych a IMIS	"	3
B. Czy WBD jest uniwersalnym rozwiązaniem problemów związanych z danymi?	"	3
C. Różnica między WBD a innym oprogramowaniem służącym do operowania danymi	"	4
D. Cel opracowania	"	4
II. Wspólna baza danych i jej otoczenie	"	6
A. Związek między użytkownikami i systemem WBD	"	7
B. Wymogi w zakresie elastyczności i ich skutki	"	8
C. Związek między WBD i innym oprogramo- waniem systemu	"	9
D. Ekonomia WBD	"	10
III. Generalny opis systemu WBD	"	12
A. Organizacja systemu	"	12
B. Możliwości zapytywania	"	14
C. Funkcje związane z organizacją logiczną ...	"	15
D. Funkcje związane z eksploatacją sprzętu ...	"	34
IV. Częściowe zastosowanie metod i technik WBD ...	"	40
A. Systemy operowania danymi	"	42

B. Pomocnicze procedury	str.	44
C. Stopniowa ewolucja w kierunku systemu WBD ..	"	48
V. Wnioski	"	49
Bibliografia	"	51

ZESTAWIENIE ZAŁĄCZNIKÓW

Nr zał.	T y t u ł	str.
1.	Główna część składowa systemu WBD	52
2.	Wewnętrzne związki między słownikiem danych, zbiorem danych i programem użytkowym	53
3.	Trójszościelowa, logicznie adresowana pamięć	54
4.	Przykład adresowania fizycznego	55

ZESTAWIENIE TABLIC

Nr tabl.	T y t u ł	str.
1.	Opis złożonych terminów wyceny w słowniku danych	56
2.	Opis złożonych terminów wyceny w słowniku danych i część opisowa danych programu związanego	58

S T R E S Z C Z E N I E

Podstawą każdego systemu informacyjnego jest zbiór danych niezbędny do jego funkcjonowania. Oczywiście dotyczy to także IMIS.¹

System informacyjny może tylko wtedy dostarczyć we właściwym czasie informacji dokładnych i wyczerpujących, jeżeli dane w zbiorze, będącym podstawą działania systemu, są dokładne i wyczerpujące, a sam zbiór jest zorganizowany w ten sposób, że wszystkie istotne dane mogą być w nim umieszczone i odszukane w przeciągu wymaganego okresu czasu. Dlatego też zdolność systemu informacyjnego do odpowiedniego wykonywania przewidzianych dla niego funkcji w dużym stopniu zależy od organizacji tej jego części, która operuje dysponowanymi zasobami danych i nosi nazwę systemu wspólnej bazy danych.

Niniejsze opracowanie, trzecie z kolei w kontynuowanej serii na temat wspólnej bazy danych, przedstawia wyniki niedawno zakończonej pracy Podkomitetu Specyfikacyjnego Systemu WBD², działającego w ramach Komitetu A do spraw IMIS w dziedzinie Wspólnej Bazy Danych. Wynikiem pracy tego podkomitetu jest opracowanie obejmujące generalny opis systemu oprogramowania WBD oraz rozważania na temat najważniejszych właściwości tego systemu i drogi dodatkowego tworzenia jego innych właściwości. Pokazano przy tym ich techniczną wykonalność.

¹ Przez "IMIS" będziemy w dalszym ciągu rozumieć zintegrowany system informacyjny zarządzania /Integrated Management Information System/. Spotyka się też w praktyce termin: "zintegrowany system informacyjny kierownictwa"/przyp.tłum./.

² WBD = Wspólna Baza Danych /ang.CDB = Common Data Base/. Używa się też w publikacjach krajowych terminów: baza wspólnych danych i bank danych /przyp.tłum./.

Ponieważ system WBD musi zabezpieczyć wykonalność różnorodnych funkcji dla pewnej liczby użytkowników mających różne i zmienne potrzeby, najważniejszymi wymogami stawianymi temu systemowi jest to, aby był on elastyczny, łatwy w eksploatacji i zdolny do dalszej rozbudowy. Cele te mogą być najlepiej osiągnięte wtedy /i bardzo możliwe, że tylko wtedy/, jeżeli system będzie zaprojektowany w sposób modułarny przy założeniu uniwersalności funkcji wykonywanych przez każdy moduł. Naszkicowany w tym opracowaniu system WBD opiera się na takim założeniu.

Koszt zaprojektowania, wdrożenia oraz eksploatacji pełnego systemu WBD może być uważany jako zbyt wysoki. Jednakże wiele metod i technik stosowanych w systemie może być używanych w mniejszej skali. Może to być efektywnym podejściem do osiągnięcia znacznych korzyści podczas kształtowania w tym samym czasie lepszej podstawy, na której ma być później utworzona wspólna baza danych.

W P R O W A D Z E N I E

A. WSPÓLNA BAZA DANYCH A IMIS

Podstawą każdego systemu informacyjnego jest zbiór danych niezbędny do jego funkcjonowania. Oczywiście dotyczy to także IMIS.

IMIS może tylko wtedy dostarczyć we właściwym czasie informacji dokładnych i wyczerpujących, jeżeli dane w zbiorze, będącym podstawą działania systemu, są dokładne i wyczerpujące, a sam zbiór jest zorganizowany w ten sposób, że wszystkie istotne dane mogą być w nim umieszczone i odzyskane w przeciągu wymaganego okresu czasu. Dlatego też zdolność IMIS do odpowiedniego wykonywania przewidzianych dla niego funkcji w dużym stopniu zależy od organizacji tej jego części, która operuje dysponowanymi zasobami danych i nosi nazwę systemu wspólnej bazy danych.

B. CZY WBD JEST UNIWERSALNYM ROZWIĄZANIEM PROBLEMÓW ZWIĄZANYCH Z DANymi ?

Wydaje się często, że istnieje duże i zrozumiałe pragnienie nałożenia na "system" - w tym przypadku system WBD - odpowiedzialności za zajęcie się pewnymi aspektami problemu, które są rozumiane tylko częściowo przez osobę, korzystającą z usług systemu. Systemy oparte na zastosowaniu komputerów, które mogą zastąpić rozumienie przez użytkownika problemu, są poza zasięgiem obecnego stanu umiejętności. Chociaż osiągnięto pewne wyniki z badań w dziedzinie systemów komputerowych, które działają rozumnie /to jest tak, jak gdyby rozumiały, co robią/, to jednak te wyniki nie zapewniają możliwości projektowania takiego systemu WBD, który mógłby automatycznie i łatwo zapewnić rozwiązanie "czarnej skrzynki" dla każdego problemu, związanego ze zbiorem danych, którym może rozporządzać użytkownik.

Jest rzeczą ważną uświadomienie sobie tego tkwiącego w istocie rzeczy ograniczenia w systemach komputerowych w tym celu, aby uniknąć spodziewania się rzeczy nieosiągalnych. Niezależnie od tego ograniczenia jest jednak rzeczą możliwą zbudowanie systemów zdolnych do czynnej pomocy użytkownikowi np: w zakresie lepszego rozumienia przez niego napotykanym problemów. WBD powinna zabezpieczyć tego rodzaju pomoc.

C. ROZNIKA WIĘDZY WBD A INNYM OPROGRAMOWANIEM SŁUŻĄCYM DO OPRACOWANIA DANYMI

Ostatnio można zauważyć duże zwiększenie możliwości technicznych, dopasowanych do różnorodności danych i warunków ich wejścia i wyjścia w systemach operacyjnych oraz w pakietach podprocedur i w całych systemach operowania danymi różnego rodzaju i służących różnym celom. Największą część, o ile nie wszystkie, pożądanych cech WBD można znaleźć wśród innych wcześniejszych systemów i procedur. Jednakże żaden pojedynczy system ani też pakiet nie posiada, ani też nie może być rozszerzony w ten sposób, aby mógł posiadać wszystkie niezbędne cechy charakterystyczne dla WBD. Kluczową wyróżniającą się właściwością WBD jest możliwość jej rozszerzenia przez dołączanie dalszych elementów.

D. CEL OPRACOWANIA

Niniejsze opracowanie, trzecie z kolei w kontynuowanej serii na temat wspólnej bazy danych, rozwija omawianie WRD publikowane uprzednio w ramach Europejskiego Programu Badawczego Diebold'a, a w szczególności zawarte w opracowaniach nr nr E29 i E48, do których odsyła się czytelnika.

Opracowanie niniejsze powstało jako wynik pracy Podkomitetu Specyfikacyjnego Systemu WBD, który miał za zadanie zbadanie bardziej

szczegółowo etapów projektowania i wdrażania WBD, a w szczególności etapów opracowywania projektów systemów WBD oraz opracowywania i doskonalenia oprogramowania WBD.

Zdecydowano, że końcową formą wyników osiągniętych przez podkomitet będzie szczegółowy opis systemu oprogramowania WBD. Opracowanie składa się z tego opisu oraz rozważań dotyczących właściwości tego systemu i drogi dodatkowego tworzenia innych właściwości systemu wg narastających potrzeb. Pokazano przy tym ich techniczną wykonalność.

Materiał w tym opracowaniu służy do:

- Wskazania użytkownikiem APD¹ głównych właściwości, jakich należy oczekiwać od systemów baz danych w bliskiej i niezbyt odległej przyszłości oraz
- Wskazania producentom komputerów i zespołom opracowującym oprogramowanie, jakie rodzaje cech są pożądane albo konieczne w świetle problemów, na które natrafiają użytkownicy w dziedzinie WBD.

Po scharakteryzowaniu tych cech i wpływów otoczenia w części II, w części III zostały przedstawione główne opisy systemu WBD. Część IV jest poświęcona rozważaniom na temat, w jaki sposób idee, metody i techniki, uprzednio wzięte pod uwagę, mogą być korzystnie eksploatowane bez narażenia się na wydatki związane z wprowadzeniem pełnego systemu WBD opisanego w części III. Wprowadzenie tych metod i technik w ramach odpowiedniego planu ogólnego może dać więc podstawę tworzenia w sposób ciągły WBD. Wnioski stanowią ostatnią część tego opracowania.

¹ APD - automatyczne przetwarzanie danych /przyp. tłum./

II. WSPÓLNA BAZA DANYCH I JKSJ O T O C Z E N I E

Wspólna baza danych, jak powiedziano uprzednio, odgrywa główną rolę w systemie informacyjnym /który nie musi być IMIS/, którego jest częścią. WBD znajduje się w wielo-aspektowym otoczeniu, poddana działaniu wielu różnych i często ścierających się z sobą czynników. Ta zróżnicowana natura otoczenia musi się odzwierciedlać w funkcjach i strukturze WBD.

W celu przygotowania usystematyzowanego przeglądu związków WBD z jego otoczeniem, jest rzeczą pożyteczną rozróżnienie następujących aspektów:

- Użytkownicy WBD i ich potrzeby. Stąd pochodzą możliwości i właściwości, które system powinien wykazywać.
- System WBD jest tylko częścią, aczkolwiek ważną, całego kompleksu, złożonego ze sprzętu i oprogramowania i zawierającego zmechanizowaną część systemu informacyjnego. System WBD działa więc wewnątrz większego systemu, obejmującego zarówno oprogramowanie - zawierające systemy operacyjne, kompilatory, procedury użytkowe, standardowe funkcje i procedury biblioteczne, programy użytkowe itd. - jak też i sprzęt zawierający jednostki pamięci operacyjnej, urządzenia pamięci pomocniczej różnych rodzajów, urządzenia wejściowe i wyjściowe zarówno przeznaczone do pracy jednoczesnej, jak też do pracy "off - line", kanały wejściowe i wyjściowe itp.
- System WBD i system informacyjny, którego jest częścią, działają wewnątrz większej organizacji ludzi dysponujących zasobami i prowadzących działalność ograniczoną warunkami ekonomicznymi i finansowymi. Ograniczenia te zwykle, o ile nie zawsze, wpływają na to, że projekty a wśród nich pożądane

wysiłki w dziedzinie APD nie mogą być najczęściej realizowane w całej rozciągłości.

A. ZWIĄZEK MIĘDZY UŻYTKOWNIKAMI I SYSTEMEM WBD

Użytkownicy systemu WBD mogą być, z grubsza biorąc, podzieleni na trzy kategorie:

- Programy użytkowe, wykonujące różnego rodzaju operacje przetwarzania danych, wykorzystujące dane z WBD i specjaliści, którzy je projektują i piszą,
- Pracownicy przedsiębiorstwa wykorzystujący system w określonych, powtarzających się operacjach w ramach normalnych codziennych funkcji biurowych,
- Użytkownicy potrzebujący z przyczyn, których nie można przewidzieć, chwilowego dostępu do pewnych danych znajdujących się już w systemie WBD. Takie życzenia są normalnie niepowtarzalne i nie mogą być z góry określone.

Programy użytkowe normalnie są eksploatowane często i wymagają szybkiego /w kategoriach prędkości maszynowych/ dostępu do dużej liczby danych. Z tego powodu wydajność komputera będzie miała duże znaczenie dla tej kategorii użytkowników, chociaż sprawność towarzyszących funkcji biurowych, takich jak przygotowanie danych i ich wprowadzenie, jest także bardzo ważna.

Z drugiej strony wydajność maszyny nie jest szczególnie ważna przy załatwianiu jakichś ubocznych nieczęstych zapytań przy małej liczbie danych; jest rzeczą ważną, aby system mógł się uporać całkowicie z zapytaniami natury nieprzewidzianej. Liczba użytkowników jest, ogólnie biorąc, duża i sprawne sterowanie ich jednoczesną współpracą z systemem WBD może być tylko przeprowadzone automatycznie przez sam system. Główną przyczyną tego rodzaju sterowania jest dążenie do uniemożliwienia nieupoważnionego

korzystania z danych w systemie i do kontroli błędów. Dlatego też system WBD powinien mieć możliwości dokładnej kontroli w tym zakresie. Należy jednak zwrócić uwagę na to, że ta kontrola nie jest łatwa i że system może oczywiście nie zapobiec temu, aby użytkownik zrobił pewne operacje, które są wprawdzie wykonalne ale niewłaściwe.

Pod wpływem tej potrzeby sprawdzania i kontroli powinny być zastosowane następujące zasady projektowania systemu:

- a/ zapewnienie istotnej pomocy użytkownikowi we właściwej eksploatacji systemu przez wykonywanie automatycznego sprawdzania /uzgadnianie sum ogólnych, obliczeń wynikowych itd./,
- b/ identyfikacja błędów użytkownika i
- c/ pomoc w ich poprawieniu, jeżeli tylko jest to możliwe.

Dążenie to nie powinno być interpretowane jako próba zwolnienia użytkownika z jego odpowiedzialności za rozumienie istoty własnego problemu i danych, których należy użyć w jego rozwiązaniu.

Zadaniem systemu WBD jest umożliwienie użytkownikowi w najłatwiejszy, jak tylko to jest możliwe, sposób sprawnego wykorzystania dysponowanych zbiorów danych; system ten nie zwalnia jednak użytkownika od konieczności dobrego orientowania się w możliwościach odpowiedniego wykorzystania dysponowanych zasobów danych. Próba niewłaściwego traktowania WBD z pewnością prowadzi do kłopotów.

B. WYMOGI W ZAKRESIE ELASTYCZNOŚCI I ICH SKUPKI

Jak to wynika z poprzednich ustaleń, system WBD musi zabezpieczyć dużą różnorodność żądań użytkowników, wynikających z ich potrzeb. Potrzeby te zmieniają się również z czasem. Jest przy tym rzeczą mało prawdopodobną, aby te wszystkie potrzeby mogły być przewidziane w czasie projektowania systemu. Dlatego też podstawowym celem projektu systemu WBD musi być elastyczność; musi być zawsze rzeczą możliwą dodanie lub modyfikacja cech WBD i wykonanie tego w sposób ekonomiczny.

Podiuga to za sobą albo dość dużą liczbę specjalizowanych procedur dla osiągnięcia wszystkich koniecznych cech systemu i w związku z tym dość duży personel programistów /aby można było, gdy zajdzie potrzeba, modyfikować i dodawać procedury/ albo też w innym przypadku mniejszą liczbę procedur, ale za to

bardziej uniwersalnych. Chociaż ostatecznie podejście może w wielu przypadkach dawać w wyniku nieco niższą wydajność komputera, to należy je, ogólnie biorąc, preferować. Pierwsze podejście jest bowiem bardzo drogie na skutek dużego wysiłku związanego z koniecznością ciągłego rozszerzania i utrzymania we właściwym stanie systemu i często przeszkadza w osiągnięciu spodziewanej elastyczności. Bez wątplenia jednak, jak wynika to z poprzednich rozważań, struktura systemu musi być dostatecznie modułarna, aby umożliwić łatwą modyfikację oraz nabywanie albo eliminację odpowiednich cech. W ten sposób ujęcie modułarne, dzięki któremu każdy moduł jest projektowany jako typowy, okazuje się najlepszą /o ile rzeczywiście, nie jedynie realną/ drogą postępowania. Z tego powodu takie podejście podano w zalecanym systemie opisanym w części III tego opracowania.

G. ZWIĄZEK MIĘDZY SYSTEMEM WBD I INNYM OPROGRAMOWANIEM SYSTEMU

System WBD, jak już uprzednio wspomniano, wchodzi w styczność z całą gamą innego rodzaju oprogramowania systemu APD. W rzeczywistości prawdopodobnie wszystkie używane w tym systemie programy spotykają się w większym lub mniejszym stopniu z systemem WBD. Fakt ten w połączeniu z okolicznością, że wiele funkcji WBD jest rozszerzeniem funkcji wykonywanych w sposób tradycyjny przez systemy operacyjne oznacza, że linia rozróżnienia między systemem operacyjnym a systemem WBD jest mglista, o ile wogóle istnieje.

Ponieważ większość programów, jeżeli nie wszystkie, włączając w to kompilatory itd., musi komunikować się z systemem WBD, inne oprogramowanie systemu informacyjnego musi być albo projektowane i doskonalone równoległe z WBD albo też dostosowywane do niej później.

Chociaż te modyfikacje wypływające z powyższego dostosowywania /w przeciwieństwie do rozszerzania zdolności systemu w celu wykorzystania nowych możliwości WBD/ nie są skomplikowane ani też trudne, to jednak mogą występować dość często

i charakteryzować się dużą pracochłonnością spowodowaną ich realizacją i testowaniem.

Może wystarczyć modyfikacja tylko części systemu operacyjnego związanej z danymi oraz z ich wejściem i wyjściem /do i z komputera/ z uwagi na to, że zakłada się, iż system ten steruje całym układem wejścia i wyjścia. Chociaż takie rozwiązanie może dawać w wyniku mniejsze wykorzystanie pełnych możliwości WBD, to jednak może być właściwym krokiem w procesie wdrażania systemu WBD.

Reasumując można stwierdzić, że system WBD jest integralną częścią oprogramowania systemu APD i powinien być jako taki traktowany. Inne części oprogramowania systemu APD powinny być projektowane i doskonalone w zgodności z WBD.

D. EKONOMIKA WBD

System WBD spełnia wiele różnych funkcji dla wielu użytkowników i jest w stałym użyciu. Dlatego też należy oczekiwać, że do jego działania potrzebna jest znaczna liczba środków maszynowych. Te "koszty stałe systemu" muszą być traktowane jako nieuniknione koszty WBD. Koszty te, oczywiście, mogą być zmniejszone przez odpowiednie projektowanie i stosowanie właściwej techniki eksploatacji, lecz nie mogą być całkowicie wyeliminowane.

Nie jest rzeczą możliwą odpowiedzieć na pytanie, ile mogą wynosić te koszty, w sposób uzasadniony obliczeniem lub wskaźnikowo, na skutek ograniczonego doświadczenia uzyskanego po dzień dzisiejszy w dziedzinie zintegrowanych baz danych. Prawdopodobnie wszystko, co może być teraz powiedziane na ten temat, jest to, że wg dzisiejszych norm koszty te najpewniej będą wysokie, nawet uwzględniając korzystne efekty wieloprogramowości.

Niektórzy obserwatorzy dochodzą do przekonania, że typowy zainstalowany system WBD sam może całkowicie zająć komputer o średniej pojemności pamięci operacyjnej plus stosunkowo małą ilość pamięci zewnętrznej /dyskowej itd./. Nie uwzględniono przy tym wyposażenia do przechowywania w pamięci zbioru danych czy też słownika danych, indeksów itd. związanych z WBD.

Czy powyższa krytyczna ocena jest słuszna, okaże się dopiero z upływem czasu.

Koszty projektowania, wdrażania i eksploatacji WBD są także trudne do oceny, jak też nawet tylko do wyodrębnienia ich od kosztów systemu informacyjnego, w którym system WBD działa. Zarówno te koszty jak też spodziewane korzyści zostały już przedyskutowane we wcześniejszych opracowaniach /E29 i E48/ na temat WBD. Nie bez znaczenia jest fakt, że uważa się, iż koszty projektowania w dużym stopniu /tak oceniają to sumarycznie pewne przedsiębiorstwa/ są kompensowane przez oszczędności na projektowaniu, doskonaleniu i pisaniu programów użytkowych.¹

Jest rzeczą ważną jednakże pamiętać o tym, że czynniki natury ekonomicznej, wpływające na decyzję co do wprowadzenia czy też nie wprowadzenia WBD, zmieniają się dość szybko na korzyść WBD. Środki maszynowe, których koszty wchodzi w skład dyskutowanych powyżej "kosztów stałych systemu", stają się z biegiem czasu tańsze i w miarę coraz większego postępu technicznego oraz zdobywania większego doświadczenia, koszty projektowania i doskonalenia oraz wdrażania powinny maleć. Jednocześnie wraz z tą dążnością osiągnięte korzyści stają się coraz ważniejsze i wartościowsze dla przedsiębiorstwa.

Jest rzeczą ważną uwzględnianie tych tendencji w analizie efektywności ekonomicznej WBD dla poszczególnych przedsiębiorstw, ponieważ sugerowanie się obecnymi kosztami i korzyściami zamiast faktycznymi przyszłymi kosztami i korzyściami, które mogą być otrzymane, da w przeważającej liczbie przypadków wynik niekorzystnie wypaczony. Odpowiednie rozważenie tych tendencji zmierza do przechylenia szali na korzyść zastosowania WBD.

¹ Programem użytkowym /application program/ nazywa się w tym opracowaniu program służący do wykonania określonej pracy lub rozwiązania określonego problemu dla użytkownika np. program służący do sporządzania listy płac, do sterowania zapasami magazynowymi, do sporządzania faktur itp./przytłum./.

III. GENERALNY OPIS SYSTEMU WBD

W oparciu o kompleksowy charakter wymagań i otoczenia systemu WBD zarysowanego w poprzednich rozdziałach przystępuje się w niniejszej części do nakreślenia generalnego opisu systemu WBD. Celem jest raczej opis głównych cech charakterystycznych systemu WBD, niż drobiazgowo i wyczerpująco wyszczególnienie sposobów jego projektowania.

Dotychczasowe doświadczenia z dużymi zintegrowanymi systemami baz danych są raczej skąpe. Z tego przede wszystkim względu wiele charakterystycznych cech tych systemów, które przytoczono lub zasugerowano na poprzedzających stronach, należy traktować jako dyskusyjne.

Zagadnienia z tym związane będą wówczas rozwiązane, gdy tylko zbierze się więcej doświadczeń pozwalających na wyrobienie sobie zdania na temat ich możliwości zarówno pożądaných, jak też i wykonalnych z punktu widzenia technicznego i ekonomicznego. Opis WBD w tym opracowaniu dlatego powinien być traktowany raczej jako propozycje w zakresie cech charakterystycznych możliwego do wykonania systemu, niż jako obowiązujące założenia dla systemu WBD wogóle.

A. ORGANIZACJA SYSTEMU

System WBD, przeznaczony do wprowadzenia, jest z konieczności stosunkowo dużym systemem wymagającym pewnej liczby ludzi zaangażowanych w opracowywanie szczegółowych projektów, oprogramowaniu systemu i realizacji procesu wdrażania. Dlatego też ogólna struktura systemu musi umożliwiać łatwe i sprawne kierownictwo pracą tych ludzi.

Jak już wspomniano wcześniej, system musi być rozszerzalny i elastyczny. Musi być rzeczą możliwą zarówno modyfikowanie i dodawanie, jak też eliminowanie funkcji i możliwości systemu bez potrzeby dokonywania zbyt dużego zamieszania

w pozostałych częściach systemu.

Wyżej podane cele mogą być w najlepszy sposób osiągnięte wtedy /i prawdopodobnie tylko wtedy/, jeżeli zastosuje się w wysokim stopniu rozwinięte podejście modularne. System powinien być oparty na stosunkowo niewielkiej liczbie prostych i jednocześnie uniwersalnych funkcji, z których, jeśli potrzeba, można składować szeroki wachlarz bardziej złożonych funkcji systemu.

Niniejsze opracowanie zostało wykonane przy uwzględnieniu potrzeb i środków, które będą w dyspozycji w najbliższej przyszłości /1 - 3 lat/. Podejście to jednak nie wykluczyło wzięcia pod uwagę także, opartych na pożądanym lecz obecnie nie będących w dyspozycji właściwościach sprzętu, cech systemu istotnych dla operacji związanych z wykorzystaniem bazy danych; i faktycznie pewne kategorie poświadczonych cech sprzętu zostały założone. Do nich np. zalicza się możliwość wykonywania przez sprzęt zorientowanych tablicowo operacji konwersji adresu logicznego na fizyczny itd. Pamięci adresowane treścią wewnętrzną¹ lub ich równoważniki podobne do stojących już do dyspozycji takich równoważników wchodzących w skład systemów pamięci paginowanych mogą zapewnić takie możliwości. Inna kategoria poświadczonych cech sprzętu umożliwia wykonanie funkcji przetwarzania danych przedstawionych w postaci wykazów /list/. Funkcje te są niezbędne zarówno do przeprowadzania niektórych operacji związanych ze słownikiem danych, jak też do umożliwienia zastosowania niektórych metod w zakresie struktury i powiązania danych.

Główną część składową systemu WBD pokazano na zał. 1. Blok "A" dotyczy fizycznego przechowywania i odszukiwania danych, konwersji adresu logicznego na fizyczny i przesyłania

¹ Pamięci te nazywane są też pamięciami adresowanymi zawartością lub pamięciami asocjacyjnymi.

danych między różnymi urządzeniami fizycznymi. Blok "B" obejmuje funkcje związane z logiczną organizacją danych w WBD. Słownik danych, przechowywany na tym samym zestawie urządzeń sprzętu, jak i wszystkie inne dane w systemie, stanowi podstawowy element sterowania. Blok "C", który działa w zasadzie tak samo, jak normalny program użytkowy, zapewnia możliwość stawiania zapytań o charakterze ogólnym.

B. MOŻLIWOŚCI ZAPYTYWANIA

Program "zapytań" WBD¹ /blok C na zał. 1/ zapewnia możliwość stawiania zapytań charakteru ogólnego w stosunku do WBD. Program jest wykonywany głównie w celu dokonania zapytań ad hoc typu nieprzewidzianego i normalnie niepowtarzalnego.

Program P - O otrzymuje od użytkowników systemu WBD problemy do sformułowania w trybie przetwarzania wielo - lub jedno-programowego. Interpretuje on te zapytania, tłumaczy je na odpowiednie rozkazy skierowane do bloku B, otrzymuje i odpowiednio tworzy układ odpowiedzi oraz dostarcza ją użytkownikowi.²

Projektowanie tego programu zapytaniowego powinno tak się odbywać, aby konieczny udział użytkownika w mechanizmie działania bazy danych był sprowadzony do absolutnego minimum. Wygoda użytkownika, a nie efektywność systemu, powinna być nadrzędnym celem projektowania; jeżeli wydajność systemu jest ważna dla pewnych typów problemów, wtedy wskazane jest zastosowanie

¹ Program "zapytań" WBD stanowi dosłowne tłumaczenie z języka angielskiego /The CDB interrogation program/. Właściwsze powinno być sformułowanie. Program "pytanie - odpowiedź" /w skrócie program P - O/ Z tego powodu wprowadzono tę zmianę w dalszym ciągu tekstu i w zał. nr 1 /prep.red./.

² Angielskie terminy: "formats", "formatting" oraz "format" zostały przetłumaczone w tym opracowaniu jako: "tworzy układ", "tworzenie układu" oraz "układ". W literaturze krajowej termin "format" tłumaczy się też jako "postać", "obraz" lub po prostu jak "format".

wanie w tym celu oddzielnego programu użytkowego.

Program P - O odniesiony do WBD opiera się na szerokim zakresie na słowniku danych w procesie interpretowania problemów stawianych przez użytkownika, w udzielaniu aktywnej pomocy użytkownikowi przy sformułowaniu przez niego właściwych z punktu widzenia merytorycznego zapytań oraz w automatycznym formatowaniu i identyfikowaniu danych w odpowiedzi udzielonej użytkownikowi.

G. FUNKCJE ZWIĄZANE Z ORGANIZACJĄ LOGICZNĄ

Część schematu, pokazana jako blok "B" na zał. 1, jest logicznym ośrodkiem systemu WBD. Jest ona zaangażowana np. w zapewnianiu i sterowaniu dostępem do każdej poszczególnej grupy danych, agendach w każdym przypadku z jej określoną strukturą logiczną. Zawarte są w niej środki do konwersji struktur, form i innych cech danych między wymienionymi pozycjami znajdującymi się w programach użytkowych, a pozycjami przechowywanymi w samej bazie danych; np. jeżeli program użytkowy śledzi pojedynczej numerycznej pozycji w układzie 6 cyfr dziesiętnych, a pozycja ta jest przechowywana wewnątrz w formie binarnej, to ten podsystem /blok "B"/ jest odpowiedzialny zarówno za wykrycie faktu, że konwersja jest konieczna, jak też za rzeczywistą jej realizację.

1. Słownik danych

Słownik danych systemu WBD zawiera opisy zbiorów danych wchodzących w skład bazy danych, logicznych zbiorów danych wchodzących w skład programów oraz wszystkich przejściowych zbiorów danych wejścia i wyjścia. Opisy te powinny dotyczyć nie tylko fizycznego rozmieszczenia danych, lecz także ich treści logicznej, stanu zabezpieczenia itd. Informacje te muszą być wykorzystywane, kiedy tylko dane są przenoszone z jednego zbioru do drugiego dla zabezpieczenia integral-

ności systemu EBD. W ten sposób może być dokonywane przeformowanie zbiorów przez odniesienie się do tabel słownika bazy danych w czasie bieżącym, używając techniki interpretacyjnej. Jednakże na skutek złożoności tych tabel może okazać się rzeszą lepszą użycie odpowiednich tabel do zestawiania modułów programów typu konwersji nośników informacji. Podejście to może być optymalne w przypadkach, gdy nie ma w ogóle potrzeby lub też zachodzi niewielka potrzeba zmiany formatu danych. Dlatego też zwykle się w tych przypadkach dyskutować zagadnienia na przykładach. W każdym razie nie należy wyciągać wniosków, że metody interpretacyjne są niewłaściwe w każdych okolicznościach.

Nie zamierzano w tym opracowaniu opisywać szczegółowo metody tworzenia słownika, ponieważ metoda ta jest w wysokim stopniu zależna od języków zastosowanych do programowania bazy danych. Przy opracowywaniu koncepcji słownika można posłużyć się informacjami, podanymi w artykule P.J.H.King'a, *Computer Journal*, t.12, 1969, str. 6-9.

a. Typy struktur danych

1/ Proste pola danych

Podstawowymi elementami strukturalnymi bazy danych są proste /elementarne/ pola danych takie jak "DZIEŃ TYGODNIA", "EGZEMPLARZ SPRZEDANY", "NAZWISKO" itd. Elementy te są s natury ogólne. Sposób ich zastosowania jednak zależy od zbiorów i rekordów, które je zawierają. Definiując te pojęcia stwierdza się, że proste /elementarne/ pola danych nie mogą już być dalej dzielone.

2/ Grupy pól

Pola danych tworzą grupy. Niektóre z nich takie, jak: "ADRES" lub "DATA" /tj. "DZIEŃ,/MIESIĄC/ROK"/ są bardziej

szeroko używane. Grupy mogą zawierać dalsze grupy w sposób powtarzalny. Rekordy, zbiory i szeregi zbiorów są specjalnymi przypadkami grup złożonych. Taki system jest z pewnością o wiele więcej elastyczny i łatwiejszy w użyciu, niż powszechnie spotykane podejście, polegające na posiadaniu oddzielnych procedur i podręcznych systemów organizowania grup danych na różnych poziomach. Pola, które występują w więcej niż jednej grupie, powinny być wyróżniane nazwą właściwej grupy, jak to jest stosowane w przetwarzającej części istniejących, wysoko rozwiniętych, języków ekonomicznych.

3/ Funkcje

Istniejące języki, takie jak COBOL, posiadają już ograniczone ułatwienia dotyczące funkcji. Dzięki temu programista może napisać "IF GROCER /tj. WŁAŚCICIEL SKLEPU SPOŻYWCZEGO/ zamiast "IF CUSTOMER - TYPE = 27 /tj. TYP ODBIORCY = 27/. Takie ułatwienia powinny być rozpowszechniane. Np. "IF MONDAY /tj. PONIEDZIAŁEK/" odnosi się do pola: "DAY - OF - WEEK /tj. DZIEŃ TYGODNIA/", jeżeli takie pole jest do dyspozycji. W braku natomiast takiego pola stosowana jest procedura obliczania określonego dnia tygodnia na podstawie grupy "DATE /tj. DATA/".

4/ Systemowe jednostki danych

W uzupełnieniu do wyżej podanych elementów strukturalnych danych należy omówić systemowe jednostki danych, których celem jest ukierunkowanie oprogramowania bazy danych. Takimi jednostkami np. są dane określające długość pola i rekordu, liczby powtórzeń pól, wskaźniki systemowe różnych typów itd. Jednostki te zwykle nie są zależne od użytkownika czy programisty. Jednakże ich opisy powinny być, w razie potrzeby, do dyspozycji, oraz powinny istnieć informacje, w jaki sposób się je stosuje.

5/ Związki między prostymi polami danych a grupami

Uzupełniając wyżej podane rozważania na temat form podstawowych jednostek danych, należy stwierdzić, że dla logicznej struktury danych ważne są związki między tymi elementami, bazujące często na wielorakich kryteriach. Związki te przybierają często formę pierścieni lub łańcuchów /tworzonych przez zastosowanie wskaźników/ lub zawarte są w indeksach. Są to niektóre tylko z bardziej znanych przykładów.

Chociaż związki tego rodzaju nie są w gruncie rzeczy niczym więcej, niż związkami w grupach danych specjalnego typu, jednakże formy, które one przybierają, mogą zaciemnić ten fakt. Dlatego też uważa się za wskazane jasno to podkreślić.

b. Zawartość pozycji słownika

1/ Nazwa systemowa

Wszystkie wyróżnione pola danych, grupa danych itd. powinny mieć odrębne nazwy w celu ich zidentyfikowania w słowniku danych. O ile to jest możliwe, nazwy te powinny być nazwami używanymi zarówno w programach użytkowych, jak też w procedurach "zapytaniowych". Dlatego też należy określić nazwy, będące synonimami, albo w sposób uniwersalny /gdy synonim znajduje się przy opisie jednostki danych/ albo też lokalnie wewnątrz określonej grupy danych.

2/ T y p

Hasło¹ słownikowe może odnosić się do pojedynczego pola, grupy, funkcji, systemowej jednostki danych, pierścienia, indeksu itd. Ponadto należy zaznaczyć, że elementy poszczegól-

¹ Hasło słownikowe lub pozycja w słowniku /ang. dictionary entry /przyp.tłum./.

nych nazw systemowych zmieniają się zależnie od tego, czy wejście następuje z kart, taśmy papierowej czy też bezpośrednio z urządzenia końcowego, czy wejście jest z drukarki /zwyczajny użytek np. jako układów usterkowych/ itd., a dalej zależnie od tego, czy są one przechowywane w bazie danych, czy też tylko w pamięci operacyjnej. W wielu przypadkach jest kilka pozycji dla tej samej nazwy danych, lecz różnych typów. A zatem "DZIEŃ TYGODNIA" ma prawdopodobnie zarówno pozycje typu wejściowego, wyjściowego i wewnętrznego jak też jedno albo więcej wyznień funkcjonalnych.

3/ Układ danych

Technika opisu wewnętrznego tworzenia układu jednostek danych jest dobrze rozwinięta /np. "PICTURE - tj. OBRAZ" w COBOL/ i nie będziemy jej tutaj dalej omawiali. Jednakże sposób, w jaki dane są przechowywane, może wpłynąć na ogólną efektywność systemu i pożądanym jest kilka uwag na ten temat.

Wiele danych w bazie danych jest alfanumerycznych. Biorąc ogólnie, jest to niewygodne, w pewnych przypadkach błędy ortograficzne mogą stanowić już pewien problem. To zostało łatwo zrozumiane. Koszty przetwarzania, pomimo dużej ilości danych, są często niskie dzięki stałej długości jednostek danych. Jednakże stała długość jednostek danych może powodować zarówno nieelastyczność na skutek dowolnego przyjmowania granic tej długości, jak też niesprawność wynikającą z zarezerwowania lecz nie zajęcia miejsca przez jednostki danych i z tych powodów może być przyjmowany układ o zmiennej długości pomimo wyższych kosztów przetwarzania.

W pewnych przypadkach nazwy alfanumeryczne mogą być przechowywane w zbiorach w formie zakodowanej, np. bardzo proste jest zakodowanie cech: "RODZAJ MĘSKI" lub "RODZAJ ŻEŃSKI" przez "1" lub "0". Jednakże użycie bardzo skomplikowanego kodowania jest często nieopłacalne, zwłaszcza gdy konwersja jest wykonywana przez komputer. W tych przypadkach rozmiar

tabel do konwersji i związanych z nią procedur może stać się nadmierny. Należy wówczas znaleźć równowagę pomiędzy minimalnym rozmiarem zbiorów, minimalnym rozmiarem programów dla wejścia/wyjścia oraz czasem przetwarzania. Użycie personelu biurowego do kodowania powoduje wzrost możliwości popełnienia błędów oraz powstanie związanych z tym problemów szkolenia. Dlatego też należy się zastanowić, aby kodowanie, o ile ma miejsce, było wykonywane automatycznie przez system. Inną z kolei wadą użycia symboli jest to, że mają one tendencje do odzwierciedlania pewnych związków logicznych występujących w pierwszych zastosowaniach systemu i mogą szybko się zdezaktualizować.

Liczby dziesiętne mają wiele cech właściwych znakom alfanumerycznym. Z powodu koniecznych nakładów czasu związanych z podstawową konwersją przy wejściu i wyjściu może często okazać się szybsze przeprowadzenie obliczeń posługując się raczej liczbami kodowanymi dziesiętnie niż binarnymi. W pewnych maszynach, w których liczby dziesiętne wyrażone są w bajtach¹, a liczby binarne znajdują się w słowach, operowanie liczbami kodowanymi dziesiętnie jest jeszcze korzystniejsze. Zagadnienie, czy dane należy przechowywać w postaci liczb kodowanych dziesiętnie czy też binarnych, należy w dużym stopniu od sprzętu. Użytkownik nie powinien wchodzić w zasadzie w te szczegóły.

Zmienny przecinek jest rzadko używany w większości bas danych ekonomicznych. Należy ze specjalną troskliwością traktować jednostki danych o niesnormalizowanej podstawie, szczególnie tam, gdzie odgrywają dużą rolę. A takie pola jak "DATA" powinny być przechowywane w systemie w formie snormalizowanej.

¹ bajt = ang. "byte" /przyp.tłum./.

4/ Struktury grupowe

Pola danych mogą być łączone razem w grupy, np.:

"CREDIT - DETAIL /tj. ELEMENT KREDYTU/ IS CUSTOMER NUMBER /tj. NUMER KLIENTA/", CREDIT - CODE /tj. SYMBOL KREDYTU/, CURRENT - BALANCE /tj. BIEŻĄCY BILANS/".

Grupy mogą same łączyć się w większe grupy w sposób powtarzalny; następujący przykład może być właściwą definicją grupy:

"INVOICE - SUITE /tj. ZESPOŁ DOTYCZĄCY FAKTUROWANIA/ IS CUSTOMER /tj. KLIENTOWSKI/ - PRICE - FILE /CENNIK/, ORDER FILE /tj. KARTOTEKA ZAMÓWIEN/, INVOICE - FILE /tj. KARTOTEKA FAKTUR/, ACCOUNT - FILE /tj. KARTOTEKA RACHUNKÓW/, ACCUMULATED-SALES - FILE /tj. KARTOTEKA ZBIORCZA SPRZEDAŻY/".

Najprostszy rodzaj grupy odpowiada rekordowi stałych pól składającemu się z szeregu oddzielnych pól danych. Jednakże wiele rekordów zawiera pewną liczbę powtarzających się pól. W wysoko rozwiniętych językach ekonomicznych określa się rzeczywistą lub maksymalną liczbę powtórzeń.

Jest też rzeczą bardzo pożyteczną, aby przeprowadzić opis sytuacji i zlecić systemowi powzięcie decyzji, jaki jest najlepszy sposób zapamiętania danych. Tak więc można używać wyrażenia "ACCUMULATED - TOTAL /tj. ZBIORCZA SUMA/ IS MONTHLY - TOTAL /tj. MIESIĘCZNA SUMA/ BY MONTH, gdzie "MONTH" oznacza

wielkość mającą 12 dyskretnych¹ wartości. Rozszerzając powyższe otrzymuje się wyrażenie: "CUSTOMER - PRICE - FILE IS CUSTOMER - PRICE - RECORD /tj. KLIENTOWSKI REKORD CENNIKOWY/ BY CUSTOMER - NUMBER", gdzie "CUSTOMER - NUMBER" oznacza sześciocyfrową liczbę, nie przyjmującą wszystkich wartości. W pierwszym przypadku dane mogą być traktowane jako 12 stałych zaindeksowanych pól o nazwie "MONTH" /tj. miesiąc/, podczas gdy w drugim przypadku dane można traktować jako numery rekordów o zmiennej długości w zbiorze rekordów: "CUSTOMER - NUMBER".

W zbiorach złożonych dane mogą być przechowywane w różnych układach strukturalnych zależnie od treści danych.

¹ dyskretny = nieciągły /ang. discrete /przyp.tłum./.

Wprowadza się w tym przypadku symbole literowe¹ w celu określenia "typu rekordu". I tak, jeżeli zamówienia przedsiębiorstw mają różny układ dla dostaw luzem i w opakowaniu, wtedy mogą one być przedstawiane jako: "ORDERS - DETAIL /tj. ELEMENT ZAMÓWIENIA/ IS CUSTOMER - NUMBER, DATE, BRAND - CODE /tj. SYMBOL ZNAKU FIRMOWEGO/, /"B", QUANTITY - IN - TONS /tj. ILOŚĆ W TONACH/. OR "P", NO-OF - CASES /tj. NIE MA POTRZEBY/, PACK - IN-CASE SIZE OPAKOWANIE W RAZIE POTRZEBY PACK-SIZE/tj. WYMIAR OPAKOWANIA/, gdzie "B" i "P" są wskaźnikami literowymi. Biorąc alternatywnie, "B" i "P" mogą być zastąpione przez funkcje "BULK /tj. LUZEM/" i "PACKED /tj. W OPAKOWANIU/", które są zdefiniowane w kategoriach dyskretnej wartości "PACKING - CODE /tj. ZBIORU SYMBOLI DOTYCZĄCYCH OPAKOWANIA/".

Ujęcie to pozwala na dużą elastyczność, ponieważ różnorodna struktury mogą znajdować się wewnątrz innych różnorodnych struktur. Pewnym problemem, spotykanym w wielu zespołach zbiorów dotyczących obliczenia sprzedaży w handlu, jest przechowywanie w kartotece umownych terminów związanych z wyceną. Tabl. 1 przedstawia sposób rozwiązania tego problemu.

Program użytkowy określa "PRICE - TERMS /tj. TERMINY ZWIĄZANE Z WYCENĄ/" tylko raz, a rozkaz "GET PRICE - TERMS" uruchamia kolejne czytanie wierszy znajdujących się w grupie. Przez użycie oddzielnych procedur dotyczących "TERM - TYPE /tj. TYP TERMINU/" można mieć do dyspozycji elastyczny i stosunkowo uniwersalny środek tworzenia i przechowywania oddzielnych terminów związanych z wyceną, zwłaszcza że mogą mieć miejsce wielokrotne wejścia wtórne dla pojedynczych pól: "BRAND", "DATE" lub "TERM - TYPE".

5/ Funkcje i ich definicje

Najprostszymi funkcjami są te, które dają nazwę wartości pola danych. Przykładem tego jest zastosowanie pola "TERM - TYPE"

¹ ang. "literals" /przyp. tłum./.

przedstawione na tabl. 1. Do określania bardziej złożonych funkcji powinno się mieć odpowiednie warunki do wprowadzania zakodowanych procedur, makrorozkazów itd. napisanych w języku zgodnym w stosunku do bazy danych. W ten sposób przy pojawieniu się w wielu zastosowaniach potrzeby określenia dnia tygodnia, można go obliczyć z daty. Inne funkcje mogą zamieniać jardy, stopy i cale na metry albo sterować tworzeniem układu na wejściu do drukarki. Funkcje te są nie tylko z natury uniwersalne, lecz także przyczyniają się do utrzymania integralności bazy danych przez eliminowanie konieczności pisania przez każdego programistę swoich własnych wersji, które mogą zawierać błędy.

6/ System i dane indeksowe

W każdej bazie danych istnieje pewna liczba indeksów, wskaźników i innych jednostek danych, których zastosowanie pozwala na szybkie operowanie danymi w odpowiednich sytuacjach. Pewne informacje, jak np. długość rekordu o zmiennej długości, mogą być traktowane jako będące całkowicie wewnętrzną sprawą struktury słownika danych i nie wymagające precyzowania ze strony użytkownika. Jednostki danych, które mają być zaindeksowane, wymagają z drugiej strony pewnej wskazówki, która by informowała, że to ma być zrobione, np.: CUSTOMER - PRICE - RECORD INDEXED BY CUSTOMER - NUMBER /tj. KLIENTOWSKI REKORD ZWIĄZANY Z WYCENĄ INDEKSOWANY NUMEREM KLIENTA/". Aby przeprowadzić takie procedury, wywoływane są /gdy potrzeba/ w oparciu o pozycje słownika danych, funkcje systemowe operujące odpowiednim zbiorem.

W rzeczywistości większość funkcji oprogramowania bazy danych jest wywoływanych przez pozycje w słowniku danych.

Dotyczy to takich zagadnień, jak uzgadnianie sum, wykonywanie kopii zbiorów, sprawdzanie uprawnień użytkownika do korzystania ze zbiorów itd.

7/ Zakres i ważność danych

Słownik danych musi zapewnić integralność bazy danych. Oznacza to, że dla każdego pola danych powinien być zachowany zakres jednostki danych lub lista dozwolonych wartości, oraz powinny być do dyspozycji funkcje sprawdzania danych takich jak "CHECKDIGIT /tj. CYFRA KONTROLNA/ - MOD - 11". Na poziomie grupy powinno być możliwe powtórne definiowanie zakresów, a także przeprowadzenie sprawdzeń krzyżowych na różnych polach wewnątrz grupy. Sprawdzenia takie powinny być wykonane na wszystkich danych wejściowych, a w miarę możliwości na wyjściu przez ostatnio wprowadzone do eksploatacji programy użytkowe. Jednakże, normalnie biorąc, pełne sprawdzenie jest zbyt pracochłonne. W warunkach przetwarzania wieloprogramowego możliwe błędy danych mogą być wychwycone przez uruchomienie programu kontrolnego na danych słownikowych dla sprawdzenia bazy danych w wolnym czasie komputera, który byłby - bez uruchomienia tego programu - nie w pełni obciążony.

8/ Dane eksploatacyjne

Należy wziąć pod uwagę rekord, którego każde pole lub grupa bierze udział w eksploatacji. Rekord ten powinien zawierać wskaźniki do odpowiednich zbiorów, indeksów lub łańcuchów. Jeśli rekord ten będzie zawierał dane statystyczne dotyczące rozmiarów zbioru, częstotliwości eksploatacji itd., wtedy może być to wykorzystane do optymalizacji organizacji bazy danych i do oceny czasów udzielania odpowiedzi na postawione poszczególne zapytania.

9/ Definicja systemu

Słownik danych powinien pozwalać na dołączanie uwag

wyjaśniających¹, o ile sądzisz tego potrzeba. Poza tym powinno być możliwe przechowywanie w języku naturalnym definicji pól, grup osy też funkcji, aby ułatwić automatyczne dokumentowanie.

e. Korzystanie ze słownika danych

Program użytkowy, realizujący proces czytania /lub pisania/ danych "prosi" system bazy danych o przeprowadzenie przesłania, podając właściwy adres nominalny /nazwę danych lub adres logiczny/ jako parametr. Oprogramowanie dokonuje konwersji podanego adresu na rzeczywisty adres w sprzęcie i wczytuje blok danych do pamięci. Zadaniem słownika danych jest przypilnowanie, aby odpowiednie części danych były umieszczone w sposób zabezpieczający dostępność dla programu. Program ten może także przeprowadzić testowanie potrzebne do utrzymania integralności bazy danych i poinformować o istniejących brakach. To można najlepiej wykonać przez porównanie formatów zbioru logicznego, wymaganego przez program, ze zbiorem fizycznym, utworzonym przez urządzenie sprzętu /zob. następny podrozdział III o 2/ oraz przez utworzenie procedury konwersji nośników danych wychodząc z pozycji słownika danych.

Najprostszy przypadek zachodzi, gdy rekordy logiczne, śladane przez program, są identyczne w układzie i zawartości, z rekordami fizycznymi w bazie danych. Wszystko, co potrzeba, należy wykonać w celu odpowiedniego zapewnienia ochrony, gdy zbiór został otwarty, oraz w celu przygotowania indywidualnych rekordów do przeniesienia do programu użytkowego jako pojedynczych jednostek. Przeniesienie to może być wykonane albo przez fizyczne skopiowanie danych albo też po prostu przez pominięcie początkowego adresu rekordu. Wybór zależy jest od tego, czy odpowiedni obszar wejściowy/wyjściowy przeznaczony jest dla bazy danych, czy też dla programu użytkowego. Taka sama procedura może być zastosowana, gdy rekordy programu są przestym

¹tj. komentarza /przyp. tłum./

podkładem rekordu bazy danych.

Rozszerzenie zbiorów zawierających pewną liczbę różnego rodzaju rekordów jest możliwe. Przypuśćmy, że dla pewnego celu różne rodzaje tych rekordów są wykorzystywane oddzielnie i mogą zajmować tę samą część pamięci. W tym przypadku logiczny zbiór programu będzie zawierał "grupy A albo B albo C". W możliwości alternatywnej rekordy wszystkich trzech typów mogą być wykorzystywane jednocześnie i wtedy logiczny zbiór będzie zawierał "grupę złożoną z grup A, B i C". W tym przypadku kodowanie, polegające na porównaniu zgodnej ze słownikiem danych definicji zbiorów logicznych ze zbiorami fizycznymi bazy danych, powoduje wczytanie i skopiowanie rekordów dowolnego typu w odpowiednich miejscach programu użytkowego.

W ogóle zbiory programowe mają stały układ. Natomiast baza danych jest organizowana w wielu przypadkach w bardziej swartym układzie. W tych ostatnich przypadkach należy dane "rozpakować" do formatu stałego przed ich użyciem oraz zastosować proces odwrotny, gdy chce się zapamiętać wyniki w bazie danych. Operacje tę można dobrze przeprowadzić używając odpowiednio przygotowanych tablic i stosując specjalne działania na urządzeniach. Szczegóły stosowanej procedury zależne są niemal całkowicie od konstrukcji będącego w użyciu komputera. Zastosowanie zaś niewłaściwej techniki może spowodować dużą pracochłonność przetwarzania. Z tego też powodu nie jest rzeczą możliwą określenie warunków na przechowywanie danych w formie zwartej w bazie danych niezależnie od rodzaju komputera.

Jest rzeczą możliwą w przypadkach skrajnych, gdy logiczny zbiór programu składał się z indywidualnych pól pochodzących z dużej liczby oddzielnych rekordów bazy danych, które mogą być rozproszone w całej tej bazie. To jest oczywiście nieefektywne, jednakże jest możliwe, jeżeli zachodzi tego potrzeba. Także powinno być możliwe, aby zbiór logiczny

zawierała pola nie istniejące w rekordach bazy danych, ale takie, które mogą być z niej wyprowadzone. W tych przypadkach stworzona procedura powinna zawierać funkcje podobne do funkcji, która oblicza dzień tygodnia na podstawie pełnej daty kalendarzowej. Dodatkowe procedury zabezpieczające mogą także wchodzić w grę w tych przypadkach.

Jeżeli zbiór fizyczny jest odcytany z bazy danych, to można zwykle założyć, że jest on logicznie prawidłowo scharmonizowany zgodnie z definicją słownika. Nie można jednak takiego założenia przyjmować, jeżeli dane są przesyłane z zewnętrznego zbioru wejściowego takiego, jak czytnik kart, do zbioru, być może też wejściowego, tworzącego część bazy danych. W tych przypadkach dane nie tylko wymagają zwykle całkowitego przeformatowania, lecz także sprawdzenia, czy są zgodne z definicjami zawartymi w słowniku danych. Tak więc słownik danych musi przeprowadzać sprawdzenie kontrolne wszystkich wprowadzonych danych.

Uważa się, że tak dalece, jak to jest tylko możliwe, wszystkie ograniczenia, które mogą wpływać na integralność bazy danych, powinny zdecydowanie być zawarte w słowniku danych oraz powinny być sterowane raczej przez ten słownik, niż przez programy użytkowe.

Procedury tworzone w tym celu, aby można było umożliwić programowi użytkowemu odczytanie ze zbioru lub wpisywanie do niego danych, są często stosunkowo proste. Jednakże, jeżeli stosuje się wyjście z programu, aby poprawić zbiór szczególnie w tym przypadku, gdy jest on indeksowany, sprawy stają się nieco więcej skomplikowane. Takie funkcje, jak te, które są potrzebne do indeksowania, tworzenia łańcuchów itd. są poza zasięgiem samego słownika danych, lecz w słowniku danych powinny być zawarte wskazania, mówiące o tym, kiedy te funkcje powinny być realizowane.

2. Rola słownika danych we współdziałaniu programów użytkowych z systemem WBD

Dwa ważne problemy nasuwają się w związku z nastawieniem systemów na korzystanie z usług słownika danych. Te problemy tkwią w następujących pytaniach: w jaki sposób współdziałają powszechnie stosowane programy napisane w językach, będących na wyższym poziomie, takich jak COBOL, z opisanym tu systemem WBD? Czy słownik danych nie może zastąpić typowej części obecnie stosowanych programów /np. części opisującej dane w programie napisanym w języku COBOL tzw. "DATA DIVISION"/ i czy w związku z tym nie są konieczne modyfikacje języków programowania, a przynajmniej teorii programowania?

Odpowiedź na pierwsze pytanie była już częściowo podana w części II C: System WBD przejmuje pewne funkcje sterowania wejściem/wyjściem oraz operowania danymi, przekazywane uprzednio systemowi monitora lub programowi w kodzie maszyny. W związku z tym zmienia on cały charakter współdziałania pracującego aktualnie programu z pozostałą częścią oprogramowania systemu w takim stopniu, w jakim dotyczy to wejścia - wyjścia oraz operacji, w których bierze udział pamięć pomocnicza. Wymaga to conajmniej takiego smodyfikowania kompilatorów, aby mogły one generować odpowiednie wywołania skierowane do systemu WBD. Po tej modyfikacji programy muszą być na nowo przetłumaczone. Aby mieć korzyść z nowości nie wprowadzonych uprzednio do stosowanego języka, należy go poszerzyć w ten sposób, jak to np. zostało niedawno zaproponowane dla języka COBOL, jeżeli chodzi o zagadnienia operowania zbiorami na urządzeniach o bezpośrednim dostępie.

Jeżeli chodzi o zagadnienia zastąpienia przez słownik części /opisującej dane/ programu, to możliwa jest zarówno odpowiedź pozytywna, jak też i negatywna, zależnie od tego, w jaki sposób system WBD został założony i zaprojektowany.

Z jednej strony można przyjąć założenie, że słownik danych w systemie WBD zastąpi części opisu danych we wszystkich programach, zapewniając w efekcie programiście znaczną pomoc dzięki automatycznemu wykonaniu /i aktualizowaniu/ znacznej części opracowywanych przez niego programów. Odpowiada to w zasadzie stosowanej przez wiele grup programistów metodzie pracy /wstawianie wyperforowanych uprzednio w dużych ilościach kompletów "DATA DIVISION" w nowe programy/.

Przyjmując, że programista dobrze zna strukturę danych i że logika jego proceduralnych ustaleń jest adekwatna do tej struktury, można stwierdzić, że wszystko jest dobrze dotąd, dopóki ta struktura danych nie jest modyfikowana. Późniejsza modyfikacja struktury danych bez sprawdzania logiki programu kończy się normalnie sytuacją pełną problemów, często nieprzewidywanych. Jednakże zdolność wykonywania zmian tej właśnie natury jest jednym z czynników składających się na elastyczność, do uzyskania której jest potrzebna WBD.

Tę elastyczność, umożliwiającą wykonanie takiej modyfikacji struktury danych bez konieczności sprawdzania i ewentualnego przepisywania wszystkich programów, mających do czynienia z danymi, których ta modyfikacja dotyczy, można osiągnąć przez wymaganie, aby opis danych był w dalszym ciągu połączony z programem i aby jednocześnie uzupełniał słownik systemu WBD /zobacz zał. 2/. W ten sposób założenia, które w innych przypadkach są domniemane, zostały sformułowane wyraźnie. Zgodnie z tą częścią programu, dotyczącą danych, opisuje strukturę danych, którymi program wg przeznaczenia będzie operował lub może operować. Słownik danych w systemie WBD opisuje strukturę rzeczywiste używanych danych. System WBD jest odpowiedzialny za wykonywanie wynikających z sytuacji konwersji, gdy są one możliwe do przeprowadzenia i gdy jest zapewnione zachowanie zgodności logicznej. Gdy zgodność ta nie może być zapewniona, system musi założyć nieodzwonność badania przez człowieka

tego stanu rzeczy. Część programu, dotycząca opisu danych, nie powinna być nigdy zmieniana automatycznie, ponieważ jest ona wyrazem przyjętych w umyśle programisty założeń w czasie, gdy pisał ten program.

Praktyczne rozwiązanie problemu, łączące w swojej istocie korzyści wynikające z obu możliwości, polega na przekopiowaniu odpowiednich pozycji słownika danych WBD i wpisaniu ich do części programu, zawierającej opis danych wtedy, gdy program został napisany /lub po raz pierwszy skompilowany/. Później ta część zawierająca opis danych jest traktowana zupełnie oddzielnie od słownika danych. Gdy słownik danych jest zmieniany, to jednak część programu dotycząca opisu danych, nie powinna być zmieniana, a uprzednio opisane sprawdzenie zgodności logicznej może być automatycznie przeprowadzone przez system WBD. Tablica 2 ilustruje różnicę między słownikiem danych a pozycjami opisu danych programu na przykładzie przedstawionym uprzednio na tablicy 1.

3. Płaszczyzna styku z programami użytkowymi

System WBD, a w szczególności podsystem "Funkcje związane z logiczną organizacją danych" pokazany jako blok B w zak.1, musi odpowiadać różnorodności rozkazów. Mówiąc prościej, rozkazy te są żądaniami o dostęp do danych i o ich zapamiętanie. Badając je z większym stopniem szczegółowości, można je podzielić na kilka kategorii:

- Operacje znajdujące się w podstawowych pakietach dotyczących sterowania wejściem-wyjściem takie, jak otwarcie zbioru, zamknięcie zbioru, czytanie i pisanie danych identyfikowanych przez nazwę lub przez /logiczny/ adres itd.
- Operacje sprawdzające i lokalizujące, ściśle związane ze strukturą danych, takie jak operacje dostępu do następnego rekordu, dostępu do poprzedzającego rekordu, dostępu do pierwszego rekordu grupy podporządkowanej itd. W systemie WBD rozkazy tego typu muszą być odpowiednio wyżej rozwinięte, niż w bardziej tradycyjnych systemach na skutek większej różnorodności struktur danych - na przykład

grup rekordów w kilku różnych sekwencjach równocześnie przy zastosowaniu wielorakich łańcuchów.

- Rozkazy o charakterze pamięci adresowanej na zawartość /treść wewnętrzną/ np.: "Usyskaj dostęp do rekordu klienta dla pola adresowego równego 15 Smith Road, miasta równego Londynowi".

Z pewnością dla indeksowanych struktur danych takie rozkazy się uwzględnią; dla innych struktur są one pożyteczne przynajmniej w pewnych okolicznościach.

- Dostęp do pozycji słownika danych potrzebny jest np. przy programie P - O WBD. Dostęp ten umożliwi programowi zapewnienie aktywnej pomocy końcowemu użytkownikowi, wskazanie ważnych oświadczeń i żądań użytkownika w różnych fazach procesu współdziałania itd.
- Operacje łączenia danych są wymagane w celu zapewnienia integralności danych, w szczególności w warunkach wieloprogramowości.
- Rozkazy dotyczące ustalania w systemie wartości granicznych różnych typów; np. ustalanie wartości granicznej liczby dostępuów fizycznych, które należy wykonać, aby przeprowadzić szczegółową kontrolę itd.

We wszystkich przypadkach dostępu do danych, włączając w to odpowiednio zaangażowane części słownika danych, stosuje się oczywiście ograniczenia wynikające ze względów bezpieczeństwa /zabezpieczenia systemu przed niepożądanym dostępem/.

Łączność systemu WBD z programem P - O jest realizowana dzięki:

- danym, będącym przedmiotem zapytań,
- potwierdzeniem wykonania żądanych operacji,
- informacjom dotyczącym okoliczności powstania usterek i błędów.

4. Bezpieczeństwo

Jak to już dyskutowano wcześniej, zabezpieczenie przed nieupoważnionym dostępem lub modyfikacją WBD oparte jest na słowniku danych. W związku z tym muszą być dostarczone odpowied-

nie informacje dla słownika danych, aby zapewnić to zabezpieczenie. Należy jednak podkreślić, że żaden system nie może zagwarantować tego zabezpieczenia w przypadku naruszenia zasad bezpieczeństwa. Podczas projektowania systemu należy przeprowadzić porównanie konsekwencji wynikających z naruszenia tych zasad z kosztami zapobiegnięcia temu naruszeniu. Potencjalni "gwakociiele" tych zasad powinni być zniechęceni do ich przekraczania przez wystarczające duże trudności i koszty naruszenia tego systemu zabezpieczenia, aby nie opłacało się im zrobienie nawet próby w tym kierunku.

Zastosowanie zastrzeżonych list dostępu, odpowiednio wbudowanych w słownik danych, łącznie z hasłem lub z innego rodzaju zakodowaną identyfikacją użytkowników, zapewnia prawdopodobnie najbardziej opłacalny sposób rozwiązania tego problemu. (Choćby metoda ta nie może gwarantować ochrony przed nieupoważnionym użytkownikiem, któremu przypadkowo może się udać odkryć ważny kod identyfikacyjny, to jednak prawdopodobieństwo zajęcia takiego przypadku może być dowolnie małe dzięki zastosowaniu odpowiedniej konstrukcji systemu kodowania.

W celu sidentyfikowania sytuacji, w których użytkownik usiłuje naruszyć system ochrony przez próbowanie wielu kombinacji kodowania, aby odkryć kod właściwy, system WBD powinien prowadzić statystykę niesrealizowanych ze względów na ochronę prób dostępu do danych. System ten powinien następnie całkowicie zablokować dostęp do tych danych po z góry określonej liczbie takich usiłowań naruszenia ochrony i zameldować o tej sytuacji osobie odpowiedzialnej za ochronę. W ten sposób takie usiłowania naruszenia bezpieczeństwa mogą być sidentyfikowane wg wszelkiego prawdopodobieństwa, zanim dane zostaną udostępnione nieupoważnionemu odbiorcy.

5. Technika przetwarzania w otaczającym środowisku

bazy danych

Przetwarzanie danych w otaczającym środowisku bazy danych obejmuje:

- wyszukiwanie informacji z istniejących zbiorów,
- zapisywanie nowych danych w wyniku zastosowania programów użytkowych,
- włączanie nowych danych do bazy danych.

Patrząc na to z tego punktu widzenia, łatwo można zauważyć, że podstawowe czynności wykonywane przez wiele programów przetwarzania danych są w rzeczywistości zupełnie podobne, nawet chociażby one wydawały się często na pierwszy rzut oka jedynne w swoim rodzaju.

Podczas projektowania systemu muszą być wzięte pod uwagę dwa ważne czynniki:

1. poprawność zbiorów - wszystkie prace z zakresu przetwarzania danych muszą być powtarzalne, jeżeli należy zapewnić ochronę przed błędami spowodowanymi przez sprzęt lub oprogramowanie,
2. czas - wszystkie dane muszą być związane z datą ważności.

Poniżej podano zalecaną metodę przetwarzania danych w środowisku otaczającym bazę danych.

a. Proces przetwarzania partiami¹

1. Przyjmij dane wejściowe albo za wejście początkowe lub też za wejście do dalszego przetwarzania z innych zbiorów w systemie.
2. Utwórz z wejścia zbiór strumieniowy /albo tymczasowy/, uwzględniając w rekordzie strumieniowym miejsce na zebranie informacji z bazy danych.
3. Możliwie po sortowaniu prześlij odpowiednie informacje z bazy danych do zbioru strumieniowego, włączając w to zaktualizowane bilanse i adresy wskaźnikowe. Nie przesyłaj w tej fazie niczego do bazy danych. Weź pod uwagę to, że ten krok w pracy stanowi całkowity program. Jest rzeczą korzystną mieć w warunkach przetwarzania wieloprogramowego programy niewielkie i proste.

1 Proces przetwarzania partiami /ang. batch process/ polega na zebraniu pewnej liczby zapytań, poprawek i/albo transakcji w jedną /partię/ i przeprowadzenie jednego zbiorczego procesu przetwarzania zamiast wielu procesów pojedynczych /przyp.tłum./

4. Następnie zaktualizuj bazę danych przesuując odpowiednie dane ze zbioru strumieniowego do właściwych zbiorów bazy danych. Ponieważ jest to tylko w zasadzie operacja kopiowania, można ją zacząć od początku w dowolnym punkcie, zabezpieczając się w ten sposób przed błędami spowodowanymi przez sprzęt i oprogramowanie.

b. Proces przetwarzania bezpośredniego¹

1. utwórz rekord strumieniowy jak dla procesu przetwarzania partiami.

2. Zbierz informacje ze zbiorów bazy danych. Zaktualizuj bilanse, lecz nie aktualizuj wskaźników ani też nie zapisuj transakcji.

3. W pewnych przedziałach czasowych przeslij zbiorczy strumień do zbiorów bazy danych i zapisz transakcje.

D. FUNKCJE ZWIĄZANE Z EKSPLOATACJĄ SPRZĘTU

1. Funkcje podsystemu związanego z eksploatacją sprzętu

Funkcje tego podsystemu /"blok A" w zał. 1/ mają za zadanie zapamiętanie i zapewnienie dostępu do bloków danych na podstawie zapytania "bloku B". Każdy blok danych jest identyfikowany przez odrębny adres logiczny, który nie jest zmieniany, gdy blok danych jest przesuwany przez ten podsystem z jednego fizycznego urządzenia pamięciowego do drugiego. Podsystem ten zapewnia przetłumaczenie adresów logicznych na fizyczne.

Adresy fizyczne są ściśle wewnętrzne dla tego podsystemu; w szczególności blok "B" nie może nigdy określić ani też odkryć

¹ Proces przetwarzania bezpośredniego /ang. on-line process/ jest procesem EPD lub pracą urządzeń, realizowaną przez komputer lub bezpośrednio przezeń sterowaną /przyp. tłum./

aktualnego umiejscowienia fizycznego jakiejś danej. Ograniczenie to jest ważnym czynnikiem pozwalającym na "przesuwanie" grup danych z jednego urządzenia na drugie, gdy wskaźniki częstotliwości dostępu do różnych grup rosną i maleją lub gdy grupy danych zostają zaktywizowane lub zdezaktualizowane przez zdania "OPEN" lub "CLOSE" albo przez ich równoważniki w programach P - O.

2. Istota płaszczyzny styku między blokami "A" i "B"

Blok B przesyła rozkazy /np. zapamiętaj, prześlij itd./, adresy logiczne i bloki danych do zapamiętania do bloku A. Blok A przesyła adresy logiczne, wyszukane dane i potwierdzenie wykonania zadania względnie wiadomości o błędach do bloku B. W ten sposób blok "B" "komunikuje się" z tzw. "pamięcią nominalną" składającą się z "segmentów adresowanych bezformatowo"¹

Z drugiej strony blok A komunikuje się albo bezpośrednio z urządzeniami pamięciowymi albo przy pomocy procedur wejścia-wyjścia w systemie operacyjnym komputera. W każdym razie blok A przesyła rozkazy dotyczące czytania / pisanie, adresy fizyczne /związane ze sprzętem/ i dane do tych procedur wejścia-wyjścia lub urządzeń fizycznych. Blok A otrzymuje odczytane dane z powrotem z pamięci oraz wiadomości dotyczące potwierdzeń, błędów i stanu sprzętu.

Proponuje się, aby adresy logiczne były zmienne i nieograniczone w swojej długości w tym celu, aby dopuścić w przypadku koniecznego rozwoju dowolne przyjęcie górnej granicy. Zaleca się, aby odpowiednią jednostką bazową długości adresów logicznych był 8-bitowy bajt dla systemu WBD eksploatowanego przy zastosowaniu komputera trzeciej generacji.

¹ Patre Madnick, S.E., "Projektowanie strategii dla systemów opartych o zbiory: model robozozy".

Proponuje się także, aby długość wymienionych przez bloki "A" i "B" bloków danych była zmienna i nieograniczona. Ze względu jednak na nieuniknione "koszty stałe" systemu /zarówno jeśli chodzi o czas, jak też i o miejsce w pamięci/ związane z logicznym adresowaniem, wydaje się, że przy pomocy jednego adresu logicznego powinien być raczej zapamiętywany rekord albo większa jednostka danych, niż pojedyncze pole wewnątrz rekordu, jeżeli wydajność komputera ma być ważnym względem, brany pod uwagę podczas projektowania. Tak więc w typowych zastosowaniach długość danych zapamiętywanych przy pomocy jednego adresu logicznego powinna wahać się od stu do kilku tysięcy bajtów.

3. Struktura adresu logicznego

W poprzednich rozważaniach stwierdzono, że adresy logiczne powinny być zmienne i nieograniczone co do długości z uwagi na założenie nieograniczania rozwoju systemu. Jednocześnie oczywiście wiadomo, że ze względów ekonomicznych jest rzeczą pożądaną, aby adresy logiczne były tak krótkie, jak tylko to jest możliwe, gdy są one zapamiętywane w jednostkach strukturalnych danych /np. wskaźniki/ i w tabelach systemu. Może to być osiągnięte przez wykorzystanie zalet hierarchicznej struktury logicznej wielkości /o ile faktycznie nie wszystkich/ dużych zbiorów danych. Duży zestaw zbiorów na dowolnych urządzeniach wykazuje tendencje do podziału na grupy zbiorów, dzielonych z kolei na zbiory indywidualne, grupy rekordów, rekordy, grupy pól, pola itd. Istnieje tendencja do koncentrowania się wzajemnych związków między danymi wewnątrz tych grup hierarchicznych. Rekordy wewnątrz zbioru są przeważnie najczęściej związane z innymi rekordami tego samego zbioru; gdy istnieją związki z danymi z innych zbiorów, to najczęściej są to związki z danymi w tej samej grupie zbiorów itd. Dlatego też jest rzeczą korzystną mieć hierar-

chiczną strukturę adresowania, w której adresowanie logiczne może być przeprowadzone wewnątrz możliwie najmniejszych grup hierarchicznych, co umożliwi otrzymanie krótkich adresów logicznych i zapewni jednocześnie możliwość nieograniczonego rozwoju.

Stąd też proponuje się przyjęcie hierarchicznej struktury adresowania logicznego przy zastosowaniu adresowania kontekstowego zwanego też adresowaniem z uwzględnieniem węzła bazowego lub kontekstowego. Taką strukturę można prawdopodobnie najjaśniej zilustrować przy pomocy wykresu choinkowego /sob. zał. 3/.

W przykładzie tym adres logiczny 7 zawiera "Dane 3"; adres logiczny 7A zawiera "Dane 8 itd. W praktyce "Dane 3" mogą być pewnymi danymi dającymi się zastosować w całej strukturze pod nimi przedstawionej na wykresie choinkowym /np. pozycje słownika danych albo oszczędnie pozycje słownika danych opisujące dalej strukturę i organizację danych/. Może być w dalszym ciągu taki np. przypadek, że jednostka strukturalna danych pod dolnym adresem 7A jest pewnym zbiorem logicznym, wewnątrz którego znajduje się wiele odczytów krzyżowych /wskazników/. Adres logiczny 7A może być zatem zadeklarowany jako węzeł kontekstowy i wszystkie adresy logiczne wewnątrz tego "zbioru" mogą być interpretowane jako adresy z uwzględnieniem tego węzła kontekstowego, ohyba, że zostanie wyraźnie inaczej powiedziane. Te adresy logiczne powinny być tylko wystarczająco długie, aby wykonać adresowanie wewnątrz zbioru. Ich długość nie musi zależeć od tego, ile jest innych nieswiązanych zbiorów w systemie. Ponadto, jeżeli zbiór ma być skopiowany pod różnymi adresami logicznymi, a żadna modyfikacja wewnątrz tych adresów nie jest wymagana w międzyczasie, to przyjmuje się, że te adresy są odniesione do węzła macierzystego nowego /skopiowanego/ zbioru.

4. Rozkazy podsystemu

Rozkazy, które ten podsystem powinien móc wykonać, mają postać:

Ustaw kontekst

Parametry wejścia: adres logiczny węzła kontekstowego i /opcjonalnie/ wskazania dotyczące warunków dostępu /szybkość, bezpośredniość, sekwencyjność itd./

Wyjście: Identyfikator kontekstowy.

Funkcja: Podsystem umiejscawia skonkretyzowany węzeł danych i porównuje jego adres logiczny z utworzonym identyfikatorem kontekstowym.

Zwolnij kontekst

Parametr wejścia: Identyfikator kontekstowy.

Funkcja: Porównywanie identyfikatora kontekstowego z uprzednio skonkretyzowanym adresem logicznym jest zaniechane. Inne operacje wewnętrzne odpowiadają "zamykaniu zbioru".

Prześlij

Parametr wejścia: adres logiczny /kompletny adres logiczny lub identyfikator kontekstowy i adres logiczny odniesiony do węzła kontekstowego/.

Wyjście: Dane zapamiętane pod skonkretyzowanym adresem logicznym.

Funkcja: realizowany jest dostęp do danych pod skonkretyzowanym adresem logicznym.

Zapamiętaj

Parametry wejścia:

- a. Adres logiczny
- b. Dane, które należy zapamiętać.

Funkcja: dane są zapamiętywane pod skonkretyzowanym adresem logicznym. Jeżeli rozkaz "DEFINE /tj. OKRESL/" nie został wykonany dla tego adresu logicznego, to jest on wtedy wykonywany.

Określ

Parametry wejścia:

- a. Oceniona długość danych /opcjonalna/.
- b. Informator /opcjonalny/: "Pod", "Obok", "Po" lub "Poniżej".

c. Adres logiczny/musi być podany, jeżeli parametr b został podany; w przeciwnym przypadku może nie być podany/.
Parametr wyjścia: określony obraz logiczny /ten sam co parametr wejścia c, jeżeli "Przy" skonkretyzowano jako opoję/.

Funkcja: wprowadza ścieżki dostępu, pozycje systemu itd. dla określonego adresu logicznego i przydziela /rezerwuje/ miejsca w pamięci. Jeżeli "Obok" lub "Po" zostało skonkretyzowane, nowo określona pozycja jest lokowana fizycznie obok albo po ustalonym adresie logicznym. Jeżeli wyszczególniono "Poniżej", nowo określony adres logiczny jest ulokowany poniżej skonkretyzowanego adresu w drzewie adresowym¹ pamięci.

Zwolnij

Parametry wejścia:

- a. Informator "Pod" albo "pod² i poniżej"
- b. Adres logiczny

Określenie wyszczególnionych adresów logicznych i przydzielenie im pamięci zostają anulowane. Dane zapamiętane w miejscach ulokowania zostają usunięte.

5. Adresowanie fizyczne

Każdy węzeł pamięciowy odpowiada fizycznemu rekordowi w pamięci. Wewnątrz rekordu fizycznego jest zapamiętywany poza swymi danymi indeks albo tabela powiązań tych węzłów bezpośrednio poniżej w drzewie pamięci. Włączone są ich fizyczne adresy, np. numery zestawów dysków, numery cylindrów, numery ścieżek itd.

¹ Jest to układ adresów w kształcie drzewa /choinki/, ang. tree /przyp. tłum./.

² "Pod" należy rozumieć w sensie kontekstu: "Pod adresem", ang. "At" /przyp. tłum./.

Proponuje się jednak, aby całkowity adres fizyczny nie był zapamiętywany w tabeli, lecz tylko te części, które różnią się od adresu fizycznego węzła macierzystego.

W ten sposób adresy fizyczne są konkretyzowane w sposób kontekstowy podobnie do struktury adresowania logicznego.

W zał. 4 zilustrowano to na przykładzie.

6. Dane WBD w pamięci centralnej /operacyjnej/

W czasie działania systemu pewne dane są wczytywane do pamięci centralnej /np. ferrytowej/. Tak długo, jak tylko na to miejsce pozwala, dane te powinny być zatrzymane w pamięci, aby odbywające się w tym czasie odniesienia się do nich mogły być przeprowadzone szybciej. Zakłada się, że takie dane, są tylko chwilowe w pamięci centralnej i że tami okraczające dostęp do nich /adresy fizyczne itp. / włączają jednak na nie, jako znajdujące się na dysku, bębnie, taśmie itd. Ostatecznie o to system musi prowadzić zestawienie informujące o tych adresach, które znajdują się aktualnie w pamięci centralnej i spraczone to zestawienie przed każdą operacją przechwylenia i zapamiętania.

7. Zestawienia sterujące w podsystemie

W omawianym podsystemie muszą być także prowadzone i inne zestawienia, wliczając w to zestawienia kontekstowe, opisy urządzeń fizycznych /nie wszystkie przechowywane w pamięci centralnej/, zestawienia zainstalowanych jednostek pamięci zewnętrznej /informujących np., że zestaw dysków 421 jest aktualnie założony na urządzenie napędowe 3 połączone z urządzeniem sterującym 4 dostępnym przez kanał 2 lub kanał 3/ itd. Zestawienia te są bardzo podobne do odpowiednich zestawień w tradycyjnych systemach sterowania wejściami i wyjściami i operowania zbiorami.

IV. CZĘŚCIOWE ZASTOSOWANIE METOD I TECHNIK WBD

Jak to już było uprzednio stwierdzone w rozdziale II D, "Ekonomika WBD", można oczekiwać, że środki potrzebne do pro-

jektowania, wdrażania i eksploatacji systemu WBD są w sumie ogólnej duże. A nawet tam, gdzie wydatki byłyby większe, niż uzyskiwane korzyści, może po prostu zabraknąć środków.

Z drugiej strony w przypadku, gdy te środki są nawet w dostatecznej ilości do dyspozycji i zapadła decyzja, aby wprowadzić pełny system WBD, zwykle jest rzeczą pożądaną, aby wprowadzać go etapami, otrzymując w każdym etapie pośrednie, dające się określić, efekty.

Z powyższych rozważań wypływa problem, czy i ewentualnie w jaki sposób można w sposób korzystny zastosować pewne metody i środki techniczne, właściwe dla WBD, bez tworzenia pełnego systemu WBD podobnego do tego, który został opisany w części III. W rzeczywistości większość /o ile nie wszystkie/ koncepcji, pomysłów i metod związanych z WBD powstało na podstawie relacji istniejących w znacznie mniej obszernych systemach; Fakt ten sam przez się prowadzi do oczywistego wniosku, że częściowe zastosowanie takich metod jest wykonalne zarówno z punktu widzenia technicznego jak i ekonomicznego. Nawet jeszcze większe zastosowanie tych metod i środków technicznych może mieć miejsce w sposób efektywny przy osiąganiu poszczególnych celów, odpowiadających potrzebom specyficznych grup użytkowników w przedsiębiorstwie.

Drogi bardziej efektywnego zastosowania pewnych metod charakterystycznych dla WBD można najlepiej rozważyć, dzieląc je na dwie główne grupy:

- dla stosunkowo samodzielnych systemów o średnich i dużych rozmiarach, nazywanych najczęściej systemami operowania danymi,
- dla mniejszych programów, procedur itd. przeznaczonych do ścisłego współdziałania z innym oprogramowaniem, procedurami organizacyjnymi itd. w zakresie korzystania z urządzeń. Będziemy je traktować jako procedury pomocnicze.

A. SYSTEMY OPEROWANIA DANymi

Liczba różnych systemów operowania danymi, które zostały opublikowane i udostępnione w ciągu ostatnich kilku lat, odzwierciedla znaczną różnorodność potrzeb użytkowników i odpowiednich sposobów eksploatacji. Poniżej podano niektóre z lepiej znanych systemów.

Auerbach	- DM - 1
Burroughs	- FORGE
Computer Sciences Corp.	- COGENT III
G.E.	- IDS
IBM	- IMS
Informatics Inc	- MARK IV
System Development Corp.	- TDMS
Scientific Data Systems Corp.	- MANAGE
Siemens	- GOLEM
Univac	- IMRADS
U.S. Army /Siły Zbrojne St.Zjedn./	- RAPID
/System zaprojektowany i udoskonalony przez Control Data Corp/	
U.S. Dept. of Defense /Min.Obrony St.Zjedn./	- MIPS

Każdy z powyższych systemów został zaprojektowany przy założeniu nieco różnych celów. W rezultacie niektórzy kładą nacisk na szybkie wyszukiwanie, odbywające się często kosztem szybkości aktualizacji i wygody, podczas gdy innym zależy

przede wszystkim na aktualizacji, a przywiązują mniejszą wagę do wyszukiwania danych. Jedni są zwolennikami przerywanego sposobu realizacji wielu pojedynczych procesów przetwarzania; inni natomiast są nastawieni na ścisłą realizację procesów przetwarzania partiami. Niektórzy popierają wielokrotne zastosowania urządzeń końcowych; inni - nie. Jedni z kolei są za fizycznie sekwencyjną strukturą zbiorów; inni zaś używają metod organizacji zbiorów polegających na zastosowaniu indeksów, dostępu swobodnego, struktur łańcuchowych lub obrotkowych itd. Rozpatrując organizację danych z większym stopniem szczegółowości, należy stwierdzić, że stosowane są różne formaty elementów danych. Niektóre systemy używają liczb dziesiętnych i formatów alfanumerycznych i nie stosują formatów binarnych, podczas gdy inni stosują formaty binarne stałe i zmiennoprzecinkowe, nie używając natomiast liczb dziesiętnych itd.

Krótko mówiąc, każdy system operowania danymi stosuje tylko niektóre z obecnie znanych metod organizacji zbiorów, technik formatowania danych, sposobów działania, sposobów zapytywania, rodzajów używanych urządzeń wejścia - wyjścia itd. W rezultacie żaden system nie jest naprawdę efektywny dla wszystkich szty też nawet dla większości zastosowań. Nie należy jednak przez to rozumieć, że takie systemy nie są użyteczne; wiele spełnia postawione przed nimi zadania całkiem dobrze. A nawet kilka systemów obsługuje nadspodziewanie duży zakres zastosowań w sposób zupełnie ekonomiczny.

Należy wyróżnić kilka wspólnych cech systemów operowania danymi, zwłaszcza systemów najbardziej udanych. Cechy te scharakteryzowano poniżej:

1. W ramach ustalonego zakresu zastosowań i sposobów działania systemów operowania danymi można znaleźć koncepcję systemu ukierunkowanego na zastosowanie uniwersalne i posiadającego obszerny zespół cech automatycznego działania. W rezultacie można otrzymać elastyczny a jednocześnie łatwy w użyciu system.

2. Wszystkie systemy mają słownik danych albo jego równoważnik w takiej czy innej formie. Stopień jego skomplikowania i rozległości zmienia się od systemu do systemu zależnie od rodzaju organizacji zbiorów oraz innych właściwości systemu.

Wyżej wymienione wspólne cechy charakterystyczne stanowią zarówno praktyczne wytyczne jak też i punkt wyjścia dla projektowania i wprowadzenia systemu WBD, systemu operowania danymi lub nawet zastosowania w mniejszej skali zasad właściwych systemowi WBD.

B. POMOCNICZE PROCEDURY

W rozdziale tym zostanie przeanalizowanych kilka z wielu sposobów, przy pomocy których można w sposób korzystny zastosować metody i techniki, właściwą dla WBD, w systemach przetwarzania danych ekonomicznych przy minimalnym wysiłku potrzebnym do wdrożenia. Przykłady te nie zakładają z góry, że w systemie planuje się czy też ewentualnie wprowadza pełny system WBD taki, jaki został przedstawiony w części III tego opracowania.

Istnieje wiele różnych dróg wprowadzenia w życie tych metod przy stosunkowo małym wysiłku. Podane poniżej wyjaśnienia są próbą zachęty czytelnika do przemyślenia dróg rozwiązania pewnych swoich bardziej nabrzmiałych problemów w zakresie bazy danych; nie zamierzamy sugerować, że podane sposoby są jedynymi czy też najlepszymi sposobami eksploatacji w małej skali pomysłów i metod w zakresie WBD. Nie zamierzamy też dawać do zrozumienia, że te metody są nowe.

1. Koordynacja kompleksu danych

Ważną zasadą wypływającą ze sposobu podejścia reprezentowanego przez WBD jest zasada centralnego: sterowania, koordynacji i dokumentacji wszystkich zbiorów danych w systemie. Przez wdrożenie systemu WBD jest się zmuszonym do wprowadzenia tej zasady w życie, ponieważ bez kompletnego, poprawnego

i jasnego określenia danych w bazie danych system WBD mógłby po prostu w ogóle nie funkcjonować.

System przetwarzania danych, nie korzystający z systemu WBD, może dzięki właściwej organizacji swojego personelu także osiągnąć dobre sterowanie i koordynację swojego kompleksu danych. Może to przybrać taką formę, że jedna osoba lub pewna grupa osób jest obciążona odpowiedzialnością za prowadzenie, w sposób ujednolicony dokumentacji dotyczącej wszystkich zbiorów danych, z których korzysta /lub z których mogą korzystać w przyszłości/ więcej niż jeden program. Gdy projektowane jest nowe zastosowanie, wtedy ta osoba lub grupa przekazuje informacje i udziela konsultacji w zakresie dysponowanych danych projektantom systemów, pracującym nad nowym systemem zastosowania danych. Jeżeli grupa odpowiedzialna za bazę danych jest zgodna co do tego, że istniejące zbiory nie zaspakajają potrzeb tego zastosowania, wtedy grupa ta pomaga w projektowaniu i koordynuje tworzenie i utrzymywanie nowych kartotek.

Osoba ta lub grupa powinna być zatem odpowiedzialna za prowadzenie zarówno listy zależności dla każdego zbioru, informującej, który program, system użytkowy itd. tworzy, modyfikuje i użytkuje dany zbiór, jak też i listy dla każdego programu, która wykazuje zbiory powiązane z danym programem. Korzystając z tych informacji, można określić i zapisać skutki proponowanych zmian w zbiorze.

2. Biblioteka słownika danych

Obecnie wiele systemów, używających języka COBOL, jest zaopatrzonych w pewną liczbę kompletów uprzednio wyperforowanych kart zawierających pozycje części, poświęconej opisowi danych, dla najczęściej używanych zbiorów danych. Gdy pisze się nowy program, uprzednio wyperforowane komplety zawierające pozycje

części, dotyczącej danych¹, dla tych zbiorów związanych z programem, są po prostu wkładane do źródłowego kompletu COBOL przed kompilacją.

Zasada tkwiąca w tej procedurze może być rozszerzona i rozwinięta przez umieszczenie tych standardowych pozycji części dotyczącej danych na taśmie, dysku, bębnie lub innych podobnych urządzeniach pamięciowych, tworząc w ten sposób bibliotekę słownika danych. Programista wpisuje do swojego programu krótkie odniesienie /np. przez nazwę zbioru/ do wejścia do biblioteki słownika danych. Prawie szablonowy program wstępny wypisuje odpowiednie pozycje części, dotyczącej danych, z biblioteki i włącza je do programu COBOL przed kompilacją.

Uzupełniając włączanie pozycji części, dotyczącej danych, do programu przeznaczonego do kompilacji, program wstępny notuje fakt wpisania pozycji części, dotyczącej danych ze zbioru X do programu Y i w ten sposób tworzy automatycznie listę zależności, przedstawioną uprzednio w punkcie IV B1. Jeżeli odpowiednie pozycje części, dotyczącej danych, w bibliotece zostały zmienione w międzyczasie począwszy od ostatniej kompilacji programu, to programista jest o tej zmianie informowany automatycznie.

Inne dane statystyczne mogą być łatwo zbierane i przekazywane grupie bazy danych np. częstotliwość, z jaką programy dotyczące każdego zbioru są od nowa kompilowane. Może się zdarzyć np., że programy wiążące się ze zbiorem X są często zmieniane, ponieważ zbiór X jest w rzeczywistości dostosowany do stawianych mu wymagań i częste zmiany stosownie do programów, odnoszących się do niego, są konieczne do załatwienia problemu.

To zaś z kolei prowadzi do wniosku, że koncepcja zbioru,

¹ ang. "data division" /przyp.tłum./

a zwłaszcza jego struktura, powinna być sprawdzana i aktualizowana.

Wyżej naszkicowaną procedurę można łatwo wprowadzić, a usprawnienia w sterowaniu, dokumentacji, użytkowaniu maszyny i pracach biurowych, związanych z programowaniem, mogą być znaczne. Poza tym stwarza się lepszą podstawę do późniejszego wprowadzenia systemu WBD.

Pewnym uzupełnieniem tej koncepcji jest pomysł włączenia do biblioteki słownika danych informacji dotyczących rozmiarów zbiorów, długości sekwencji, długości łańcucha dla samej danej itd. oraz wykorzystywania tych informacji, gdy decydują one np. o sposobie wyszukiwania w tych przypadkach, gdy wybór takiego sposobu jest możliwy. Innym z kolei pomysłem, który można wprowadzić, jest automatyczne tworzenie /jeżeli tylko jest to logicznie możliwe/ programu przeformatowywania zbioru, gdy zostały zmienione dla jakiegoś zbioru pozycje słownika danych.

3. Biblioteka procedur związanych z dostępem do bazy danych

Bibliotekę można stworzyć w języku źródłowym; zawiera ona procedury dla niektórych z częściej występujących operacji dostępu do bazy danych takich, jak wyszukiwanie w indeksowanym zbiorze, swobodny dostęp na podstawie wskaźników itd. W ten sposób mogą być stosowane metody manipulacji danymi i organizowania zbiorów, które nie są bezpośrednio wspomagane przez system wejścia - wyjścia w oprogramowaniu monitora komputera i w oprogramowaniu towarzyszącym. Te nowe procedury mogą oczywiście korzystać z oprogramowania wejścia - wyjścia monitora, rozszerzając i specjalizując je dla potrzeb SEPD. Niektóre z tych procedur mogą np. posługiwać się:

1. rekordami o zmiennej długości w systemach przewidujących jako normę stosowania tylko rekordów o stałej długości,
2. strukturami drzewiastymi /np. zastosowania częściowo gwałtownie się rozszerzające/ w systemach dopuszczających tylko struktury sekwencyjne i indeksowo sekwencyjne,

3. konwersją formatów przy zetknięciu się skomplikowanej struktury zbioru z językiem programowania nie będącym w stanie dokonywania opisu ani też manipulowania tego rodzaju strukturą zbioru itd.

Tego rodzaju biblioteka procedur w zakresie dostępu nie różni się od typowych bibliotek programów i podprogramów innych systemów. Podana w tym miejscu sugestia jest tego rodzaju, aby podjęto wysiłki w celu stworzenia skoordynowanej grupy prostych w zastosowaniu lecz jednocześnie elastycznych i uniwersalnych modułów programów. Moduły te mogą w znaczny sposób rozszerzyć możliwości SEPD w zakresie organizacji, manipulowania i efektywności wykorzystania kompleksu danych, który jest do dyspozycji zarówno obecnie jak też będzie w przyszłości.

C. STOPNIOWA EWOLUCJA W KIERUNKU SYSTEMU WBD

Uprzednie wyjaśnienia w zakresie częściowego zastosowania metod i sposobów technicznych właściwych WBD wskazują na silne podobieństwo tego częściowego zastosowania do początkowych etapów zawartych w wyraźnym planie stopniowego rozwoju oprogramowania w kierunku wprowadzenia systemu WBD. I faktycznie każdy etap w takim planie stopniowego rozwoju stanowi model efektywnego zastosowania tych metod w systemie o mniejszym zakresie niż zakres pełnego systemu WBD.

Na pewnym np. etapie należy system wyposażyć dodatkowo w cechy zapewniające ochronę przed nieupoważnionym dostępem do danych i wprowadzeniem danych. Cechy te można włączyć do procedury zakładania i korzystania z biblioteki słownika danych opisaną uprzednio w punkcie IV B2.

Wysoko modularna struktura systemu WBD lub jego ewolucyjnych poprzedników umożliwia łatwe wyposażenie systemu w podobne cechy przez dodanie lub modyfikację modułu lub niewielkiej grupy modułów. Np. na dalszym etapie ewolucji systemu WBD automatyczne przesyłanie danych między jednostkami pamięci

/"wprowadzenie, zabranie"¹/, może być realizowane przez modyfikowanie dostępu do danych na najniższym poziomie i modułów pamięci oraz przez użycie dawnych rzeczywistych adresów jako nowych adresów logicznych /sob. rozdział III D/.

V. WNIOSKI

1. System wspólnej bazy danych jest istotną częścią IMIS, lecz nawet wtedy, gdy ten system nie stanowi części IMIS, można oczekiwać z jego zastosowania bardzo poważnych korzyści.
2. Koszty projektowania, wprowadzania i eksploatacji wspólnej bazy danych pozostają we właściwej proporcji do uzyskiwanych korzyści; te znacząco są wysokie. Szczególnie trudną rzeczą jest uzyskanie dobrze wykwalifikowanego personelu fachowego.
3. Korzyści uzyskiwane przez stosowanie WBD są szczególnie duże dla przedsiębiorstw i instytucji, które mają duże działy projektowania systemów i programowania i które napotykają kłopotliwe problemy łącznościowe w sferze koordynacji danych i zbiorów danych.
4. Częściowe zastosowanie właściwych dla WBD metod i sposobów technicznych na mniejszą skalę, niż to się dzieje przy korzystaniu z pełnego systemu WBD, jest realne i może być efektywną drogą otrzymania znacznych korzyści.
5. Częściowe zastosowanie indywidualnych sposobów technicznych, o ile jest wdrażane w ramach odpowiedniego planu, może stanowić lepszą podstawę dla późniejszego wprowadzenia systemu WBD w pełnym wymiarze.
6. Jednym z najważniejszych elementów systemu WBD jest słownik danych; jest on częścią centralną, wokół której tworzona jest reszta systemu.
7. Modularne podejście do projektowania systemu WBD okazuje się podejściem najlepszym dla efektywnego rozwoju i wdra-

¹ Ang. "roll-in, roll-out" /przyp.tłum./

kania elastycznego prasa w sposób dynamiczny doskonałego systemu WBD.

8. System WBD musi być elastyczny a jednocześnie łatwy do zastępowania. Cele te można prawdopodobnie najlepiej osiągnąć przez stosowanie w projektowaniu zasady jak największej uniwersalności funkcji, cech języka itd.
9. Dla specjalnej kategorii potrzeb systemy operowania danymi są całkowicie użyteczne, lecz ich rozszerzanie, zmniejszanie lub też usiłowanie soalenia z innymi systemami jest często problematyczne i w pewnych przypadkach niewykonalne.
10. Najbardziej postępowym z punktu widzenia WBD sprzętem jest sprzęt zorientowany na naprawę niedrogie pamięci masowe, tabele i indeksowanie, jak np. sprzęt zawierający pamięć adresowaną zawartością /PAZ¹/.

1 PAZ - pamięć adresowana zawartością /ang.CAM - content addressable memory/ przyp.tłum./

BIBLIOGRAFIA

Data Base Task Group /Grupa Robocza w zakresie Bazy Danych/,
Komitet "Codasyl COBOL": "Sprawozdanie dla Komitetu "Codasyl
COBOL" w zakresie rozszerzania języka COBOL "potrzebnego
do eksploatacji baz danych", styczeń 1968.

SIGBDP /ACM/ Newsletter, kwiecień 1968.

- Otrzymał uwagi w sprawie "Rozszerzania języka COBOL potrzebnego do eksploatacji baz danych". SIGBDP /ACM/
Newsletter nr 6, sierpień 1968.

Dodd G.G.: "Elementy systemów operowania danymi. Computing
Surveys /ACM/, t.1, nr 2, czerwiec 1969.

Hoffman L.J.: "Komputery a zachowanie tajemnicy. Przegląd
zagadnień". Computing Surveys /ACM/, t.1, nr 2, czerwiec 1969.

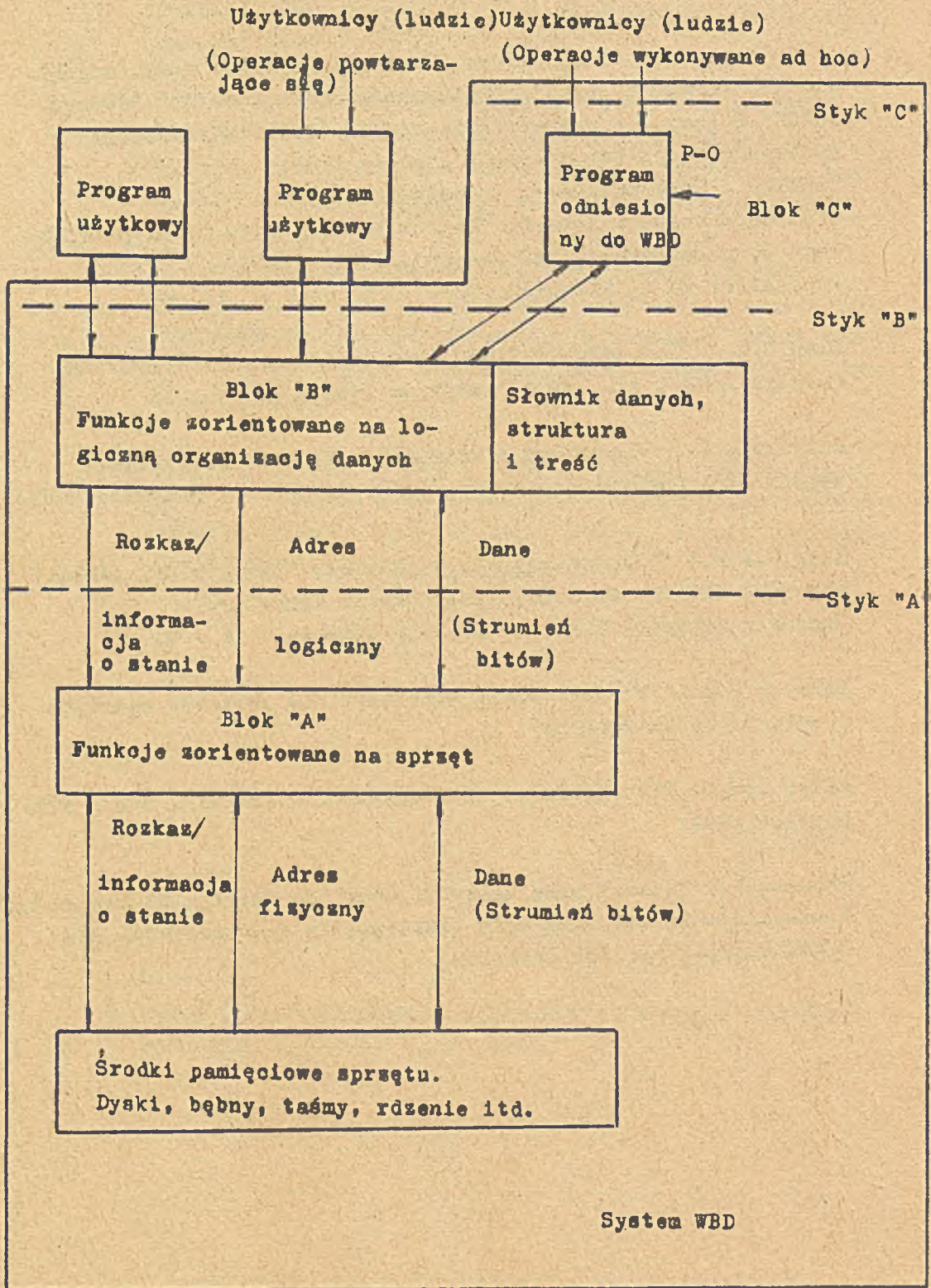
King P.J.H.: "Dokumentacja projektowania systemów. Określenie
słownika danych tworzonych przy użyciu komputera".
Computer Journal, t.12, nr 1, luty 1969, str. 6 - 9.

Madnick S.E.: "Projektowanie strategii dla systemów opartych
o zbiory. Model roboczy".

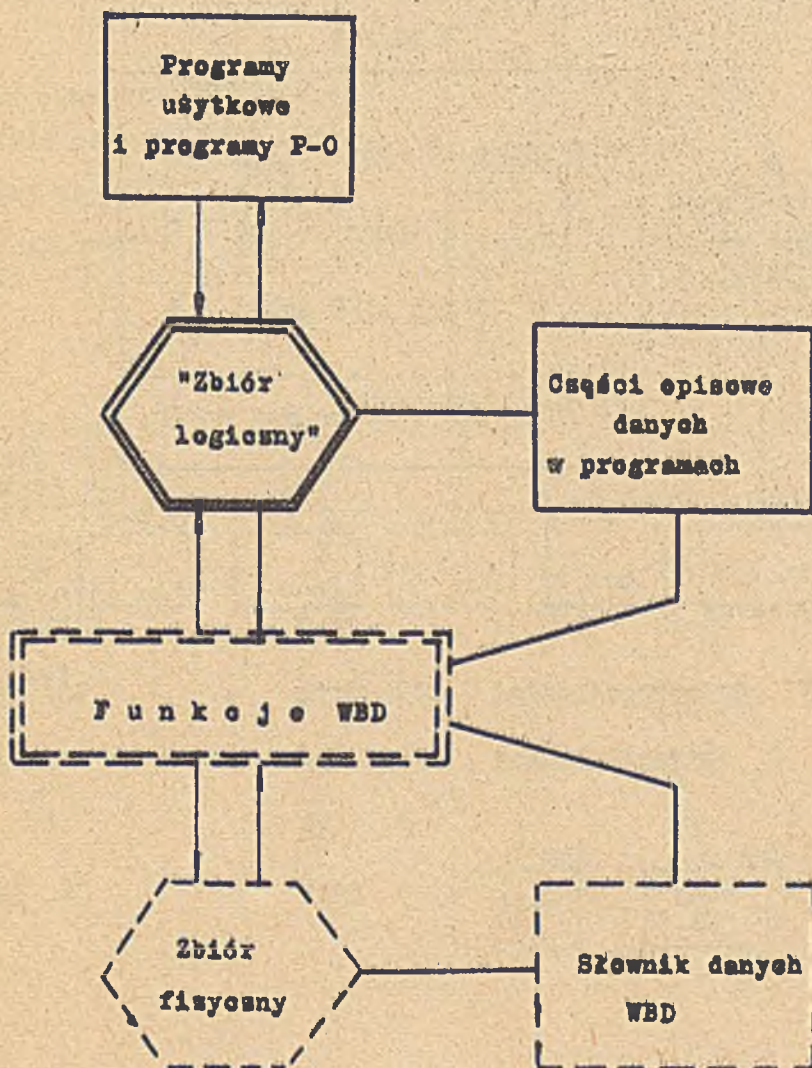
Mitre Corp.: "Przegląd systemów operowania danymi". MTP - 329,
styczeń 1969.

Thomson B.: "Czas a baza danych". Computer Weekly, 24 lipiec 1969.
W uzupełnieniu zob. rozległą bibliografię w dokumentach E29
i E48 Europ.Progr.Bad.Diehelda.

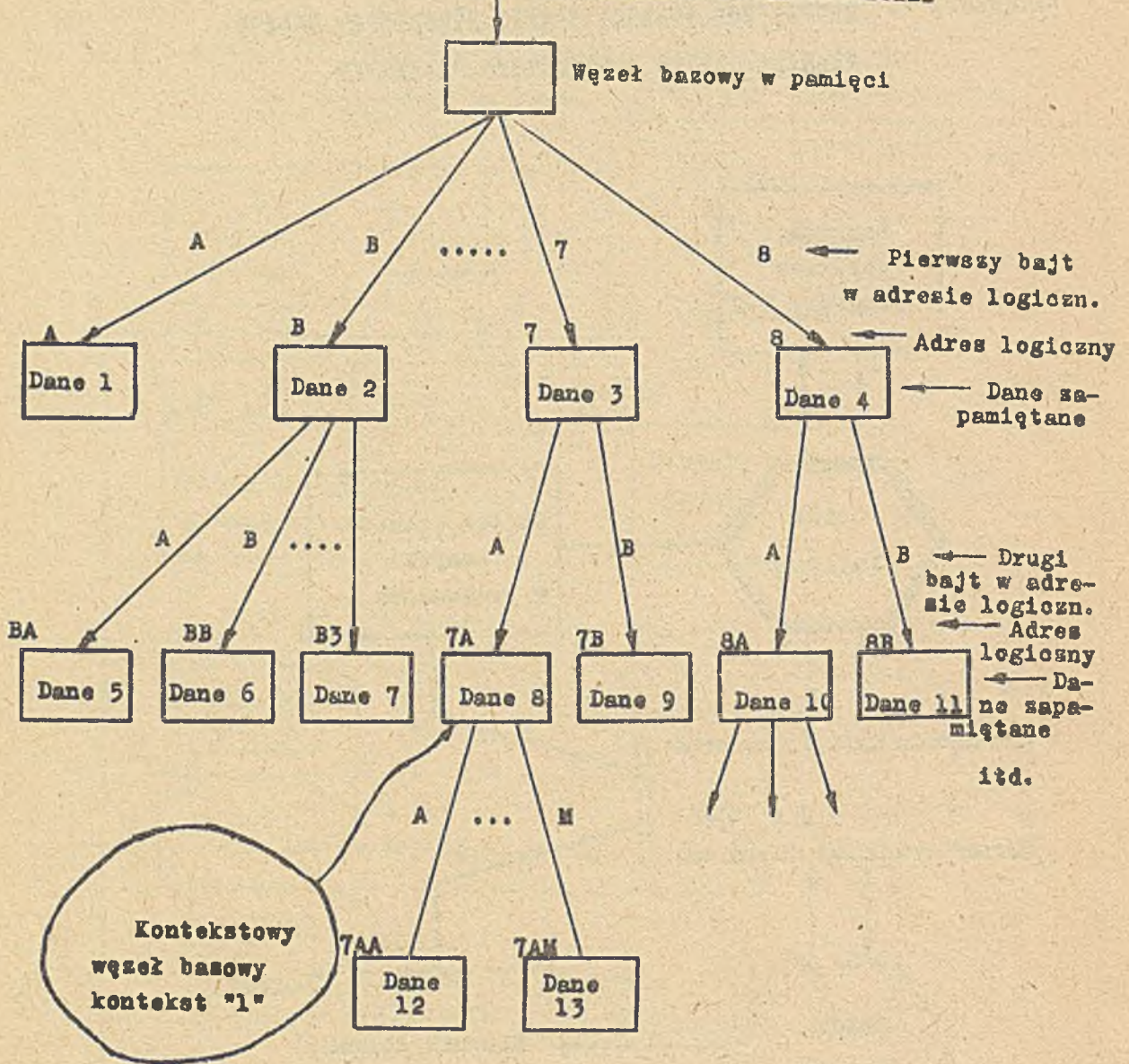
Załącznik 1. Główna część składowa systemu WBD



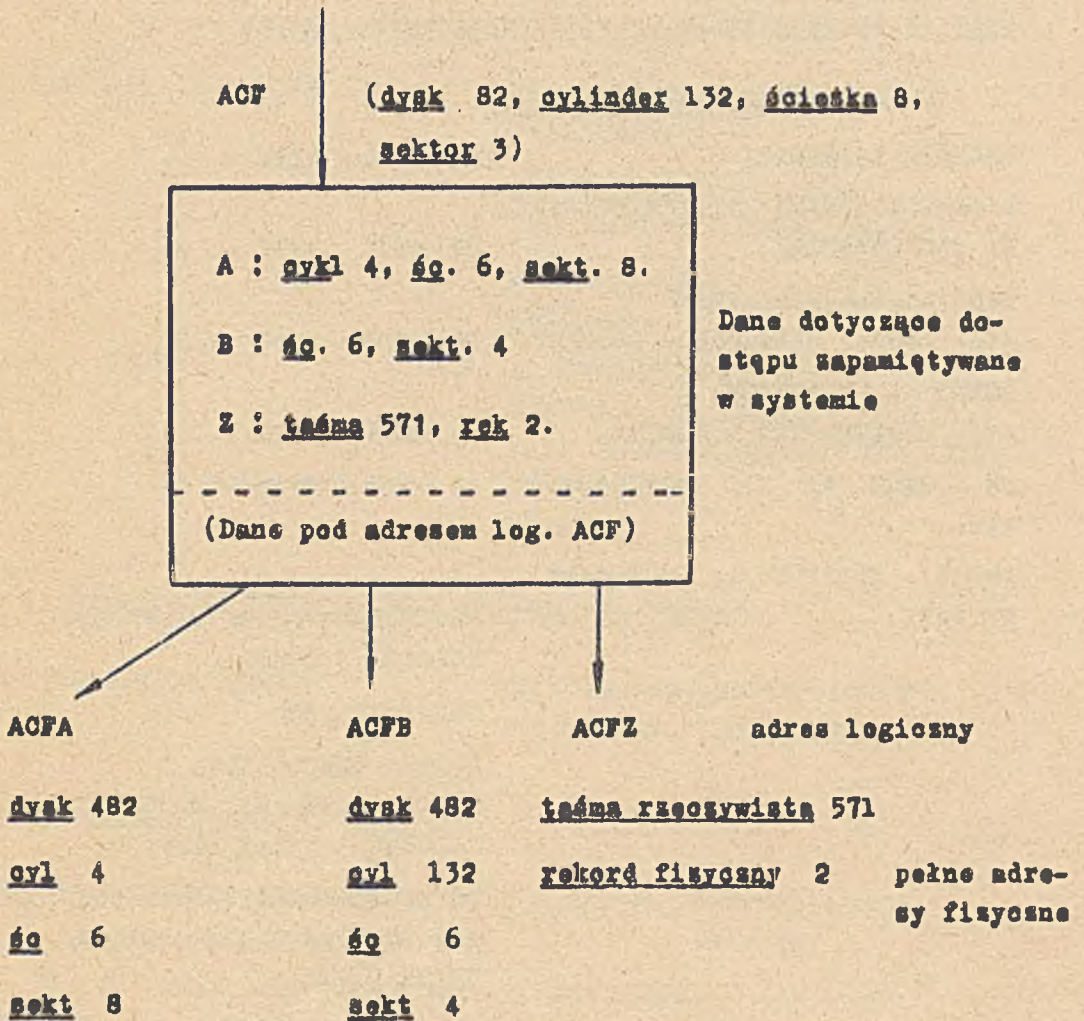
Załącznik 2. Wewnętrzne związki między słownikiem danych, zbiorem danych i programem użytkowym



Załącznik 3. Trójszczeblowa, logicznie adresowana pamięć



Załącznik 4. Przykład adresowania fizycznego



TABLICA 1

Opis złożonych terminów wyceny w słowniku danych

Określenie bazy danych

REBATE /tj. RABAT/	IS TERM - TYPE = 1
LOAD-CONCESSION /tj. ZEZWOLENIE NA ZAŁADOWANIE/	IS TERM - TYPE = 2
SPECIAL-DELIVERY-TERM /tj. TERMIN SPECJALNEJ DOSTAWY/	IS TERM - TYPE = 3
AGENT - COMMISSION /tj. POWIERZENIE W KOMIS/	IS TERM - TYPE = 4
NO - SALE /tj. NIE SPRZEDANO/ itd.	IS TERM - TYPE = 5
PRICE - DETAILS /tj. ELEMENTY WYCENY/	IS REBATE, PRICE DIFF /tj. ROŻNICA W CENIE/ OR /tj. ALBO/ LOAD - CONCESSION, LOAD - SIZE /tj. ROZMIAR ŁADUNKU/, PRICE DIFF. OR SPECIAL-DELIVERY-TERMS, NOTE /tj. POTWIERDZENIE ODBIORU/, PRICE-DIFF. OR AGENT-COMMISSION, AGENT-NUMBER /tj. NUMER PRZEDSTAWICIELA/, COMMISSION /tj. KOMIS/. OR NO-SALE, REJECTION-CODE /tj. OZNACZENIE BRAKU/ itd.

PRICE - TERM /tj. TERMIN
ZWIĄZANY Z WYCENĄ

IS BRAND /tj. ZNAK FIRMOWY/
DATE, PRICE - DETAILS.

CUSTOMER-PRICE-RECORD

IS CUSTOMER-NUMBER, ADDRESS
/tj. ADRES/,

PRICE-TERM BY/tj. PRZEZ/
/BRAND, DATE, TERM-TYPE/,

OTHER-DETAILS /tj. INNE
SZCZEGÓŁY/.

PRICE - FILE

IS CUSTOMER-PRICE-RECORD BY
CUSTOMER, BRAND-PRICE-RECORD
/tj. REKORD CENNIKOWY DLA ZNAKU
FIRMOWEGO/
BY BRAND.

Określenie programu do spersadzania faktur

CUSTOMER-PRICE-RECORD

IS CUSTOMER-NUMBER, ADDRESS,
PRICE-TERM, OTHER-DETAILS.

Ważne rozkazy wejścia

GET /tj. POBIERZ/ CUSTOMER - PRICE - RECORD /dla następnego
klienta/

GET PRICE - TERM /dla następnego układu terminów dla tego
samego klienta/.

TABLICA 2

Opis skróconych terminów wyceny w słowniku danych
i esadź opisowe danych programu swiasanego

Deszycie w słowniku danych i w esadźci opisowej danych
PROGRAMU DO SPORZADZENIA FAKTUR

REBATE	IS TERM - TYPE = 1
LOAD - CONCESSION	IS TERM - TYPE = 2
SPECIAL - DELIVERY - TERM	IS TERM - TYPE = 3
AGENT - COMMISSION	IS TERM - TYPE = 4
NO - SALE	IS TERM - TYPE = 5
itd.	
PRICE - DETAILS	IS REBATE, PRICE - DIFF. OR LOAD - CONCESSION, LOAD-SIZE, PRICE-DIFF. OR SPECIAL-DELIVERY-TERMS. NOTE, PRICE - DIFF. OR AGENT - COMMISSION, AGENT - NUMBER, COMMISSION OR NO - SALE, REJECTION-CODE itd.
PRICE - TERM	IS BRAND, DATE, PRICE-DETAILS.

Penycie tylko w słowniku danych

CUSTOMER - PRICE - RECORD	IS CUSTOMER - NUMBER, ADDRESS, PRICE - TERM BY /BRAND, DATE, TERM - TYPE/, OTHER DETAILS.
---------------------------	---

PRICE - FILE

IS CUSTOMER - PRICE - RECORD
BY CUSTOMER, BRAND - PRICE -
RECORD BY BRAND.

pozycje tylko w programie do sporządzania faktur

CUSTOMER - PRICE - RECORD

IS CUSTOMER - NUMBER, ADDRESS,
PRICE - TERM, OTHER DETAILS.

Ważne rozkazy wejścia

GET CUSTOMER - PRICE - RECORD /dla następnego klienta/

GET PRICE - TERM / dla następnego układu terminów dla
tego samego klienta/.

