R. 1877/93

5 1993

informatyka

WYDAWNICTWO SIGMA NOT

Cena 25 000 zł

informatyka

nr 5 1993
Maj
Rok wydania XXVIII

Miesięcznik
ISSN 0542-9951
INDEKS 36124

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
mgr inż. Piotr FUGLEWICZ
mgr Teresa JABŁOŃSKA
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
mgr inż. Jan RYŻKO
dr Zdzisław SZYJEWSKI

PRZEWODNICZĄCY
RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:
Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT
Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:
01-552 Warszawa,
Pl. Inwalidów 10, p. 104, 105
tel. 39-14-34

Materiałów nie zamówionych
redakcja nie zwraca

W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji lub Działu Reklamy i Marketingu 00-950 Warszawa ul. Mazowiecka 12 telefon: 27-43-66 telefaks: 19-21-87 teleks: 814877

W numerze

Strona

Linda – środowisko do przetwarzania równoległego i rozproszonego w sieciach stacji roboczych – *Grzegorz Czajkowski, Krzysztof Zieliński* 1
ECHO – komputerowy serwis informacyjny – *Danuta Bartoszek* 6
Prosta integracja środowiska sieci NetWare ze światem UNIX – *Leszek Kotzian* 14
Rekomendacje narzędzi graficznych dla środowiska APW – *Dariusz Gaik, Andrzej Przybylski* 17
Problemy projektowania systemów informacyjnych – *Jan Goliński* 19

Z kraju
The Third International Conference Information Systems Developers Workbench, Sopot 22–24 września 1992 r. – *Stanisław Wrycza, Witold Rybiński* 22
Założenie wdrożenia w RP Systemu Rozwoju Zawodowego na licencji Brytyjskiego Towarzystwa Komputerowego – *Piotr Fuglewicz, Paweł Gizbert-Studnicki* 25

Ze świata
Szybki rozwój systemów Windows – *Jan Ryżko* 27

Warunki prenumeraty na 1993 r.

Zamówienia na prenumeratę czasopism wydawanych przez Wydawnictwo SIGMA-NOT można składać w dowolnym terminie. Mogą one obejmować dowolny okres czasu, tzn. dotyczyć dowolnej liczby kolejnych zeszytów każdego czasopisma.

Zamawiający może otrzymywać zaprenumerowany przez siebie tytuł począwszy od następnego miesiąca po dokonaniu wpłaty. Zamówienia na zeszyty sprzed daty otrzymania wpłaty będą realizowane w miarę możliwości – z posiadanych zapasów magazynowych.

Warunkiem przyjęcia i realizacji zamówienia jest otrzymanie z banku potwierdzenia dokonania wpłaty przez prenumeratora. Dokument wpłaty jest równoznaczny ze złożeniem zamówienia.

Wpłat na prenumeratę można dokonywać na ogólnie dostępnych blankietach w Urzędach Pocztowych (przekazy pieniężne) lub Bankach (polecenie przelewu), przekazując środki pod adres:
Wydawnictwo SIGMA-NOT Spółka z o.o.
Zakład Kolportażu
00-950 Warszawa, skr. poczt. 1004
konto:
PBK S.A. III 0/Warszawa nr 370015-1573

Wpłaty na prenumeratę od marca br. przyjmują także wszystkie urzędy pocztowe nadawczo-odbiorcze oraz doręczyciele na terenie całego kraju

Na blankiecie wpłaty należy czytelnie podać nazwę zamawianego czasopisma, liczbę zamawianych egzemplarzy, okres prenumeraty oraz własny adres.
Na życzenie prenumeratora, zgłoszone np. telefonicznie, Zakład Kolportażu ul. Bartycka 20, 00-950 Warszawa, (telefony: 40-30-86, 40-35-89 oraz 40-00-21 wew. 249, 293, 299) wysyła specjalne blankiety zamówień wraz z aktualną listą tytułów i cennikiem czasopism.

Odbiorcy zagraniczni mogą otrzymywać czasopisma poprzez prenumeratę dewizową (wpłata dokonywana poza granicami Polski w dewizach, wg cennika dewizowego z cenami podanymi w dolarach amerykańskich) lub poprzez zamówioną w kraju prenumeratę ze zleceniem wysyłki za granicę (zamawiający podaje dokładny adres odbiorcy za granicą, dokonując równocześnie wpłaty w wysokości dwukrotnie wyższej niż cena normalnej prenumeraty krajowej).

Egzemplarze archiwalne (sprzedaż przelewowa lub za zaliczeniem pocztowym) można zamawiać pisemnie, kierując zamówienia pod adresem: Wydawnictwo SIGMA NOT, Spółka z o.o. Zakład Kolportażu, 00-716 Warszawa, ul. Bartycka 20, paw. B, tel. 40-37-31, natomiast za gotówkę można je nabyć w Klubie Prasy Technicznej w Warszawie ul. Mazowieckiej 12, tel. 26-80-17.

Istnieje możliwość zaprenumerowania 1 egz. czasopisma po cenie ulgowej przez indywidualnych członków stowarzyszeń naukowo-technicznych zrzeszonych w FSNT oraz przez uczniów zawodowych i studentów szkół wyższych. Blankiet wpłaty na prenumeratę ulgową musi być opatrzony na wszystkich odcinkach pieczęcią koła SNT lub szkoły.

W przypadku zmiany cen w okresie objętym prenumeratą Wydawnictwo zastrzega sobie prawo do wystąpienia o dopłatę różnicy cen oraz prawo do realizowania prenumeraty tylko w pełni opłaconej.
Cena jednego egzemplarza: normalna 25 000 zł, ulgowa 18 750 zł
Wartość prenumeraty w zł:
Normalna: kwartalna 75 000, półroczna 150 000, roczna 300 000
Ulgowa: kwartalna 56 250, półroczna 112 500, roczna 225 000.

Skład i druk: Drukarnia SIGMA NOT Sp. z o.o. z. 148/93

GRZEGORZ CZAJKOWSKI
KRZYSZTOF ZIELIŃSKI
Instytut Informatyki AGH
Kraków

P. 1877/93

Linda – środowisko do przetwarzania równoległego i rozproszonego w sieciach stacji roboczych

Typowa obecnie sytuacja: użytkownik chce napisać program, wykonywany na sieci stacji roboczych, lecz nie zdecydował się jeszcze na wybór konkretnego narzędzia umożliwiającego przetwarzanie rozproszone. Czasem problem rozwiązuje się sam, bo jest dostępne tylko jedno środowisko, w którym można zrealizować obliczenia sieciowe. Jednakże rosnąca liczba, różnorodność i – co równie ważne – dostępność odpowiednich systemów powoduje, że nie zawsze łatwo jest podjąć decyzję: będę programował w *X*.

Celem tego artykułu jest prezentacja bardzo interesującej możliwości – wybrania środowiska Linda[1]. Linda nie jest samodzielnym językiem programowania, lecz zbiorem kilku operacji umożliwiających koordynację i współpracę programów napisanych w już istniejącym języku – mówimy więc o C-Lindzie czy o Lindzie opartej na Moduli. Oryginalne podejście do zagadnień obliczeń rozproszonych, ogromna prostota nie ograniczająca jednak możliwości programowania, a także efektywność dostępnych komercyjnie systemów (cytat z [5] „... runing Linda on 14 Vaxes, some in California and some in New Mexico, Robert Whiteside (..) has handily beat a Cray") decydują o atrakcyjności Lindy.

Artykuł podzielono na cztery części. W części pierwszej opisano model obliczeń Lindy, druga zawiera omówienie ogólnodostępnej implementacji C-Lindy, systemu POSYBL-1.102[2]. Wybór tego systemu został podyktowany względami ekonomicznymi: sprzedawane implementacje Lindy są bardzo drogie, natomiast POSYBL-1.102 jest dostępny dla wszystkich zainteresowanych na zasadzie *public domain*. W części trzeciej pokażemy wybrane techniki programowania. Ostatnia, czwarta część zawiera opis kierunków rozwoju Lindy.

Model obliczeniowy Lindy

Linda została zaprojektowana przez Davida Gelerntera z Yale University i jest obecnie rozwijana i udoskonalana przez pracowników tej uczelni. Opracowaniem i dystrybucją komercyjnych wersji Lindy zajmuje się firma *Scientific Computing Associates, Inc.*

Z Lindą są związane dwa podstawowe pojęcia: krotka (ang. *tuple*) i przestrzeń krotek (ang. *tuple space*), nazywana skrótowo TS. Krotka jest ciągiem pól. Każde pole ma własny typ oraz ustaloną wartość lub potencjalną wartość[3]. Mówimy, że pole krotki ma wartość potencjalną, jeżeli w wyniku wykonywania przez Lindę operacji na danej krotce polu temu zostanie nadana konkretna wartość. Na przykład:

(*"first_tuple"*, 1, ? *val*)

jest krotką o trzech polach: pierwsze jest napisem *"first_tuple"*, drugie – liczbą całkowitą 1, a trzecie – zmienną *val*, wyrażeniem mającym potencjalnie jakąś wartość.

Przestrzeń krotek to dostępny dla wszystkich procesów zbiór krotek. Model Lindy jest abstrakcyjnym modelem pamięci, w którym każdy proces ma dostęp do danych umieszczonych w przestrzeni krotek. Procesy mogą także tworzyć dane bezpośrednio dostępne dla innych procesów przez umieszczenie odpowiednich krotek w TS. Nie należy jednak utożsamiać przestrzeni krotek z pamięcią dzieloną **fizycznie**. Przestrzeń krotek jest rozproszona na sieci stacji roboczych. Jeżeli interesująca dany proces krotka znajduje się fizycznie na innej stacji, to jest z niej pobierana. Powoduje to oczywiście określone narzuty czasowe, związane z koniecznością obsługi zdecentralizowanej przestrzeni krotek.

Linda umożliwia dokonywania następujących czterech operacji na TS:

- Wkładanie krotki do TS. Umożliwia to operacja *out*. Na przykład *out* (*"example"*, 4) umieści w TS krotkę (*"example"*, 4). Wartości pól krotki są obliczane przed jej włożeniem do TS. Warto zwrócić uwagę na fakt, że w TS może znajdować się więcej niż jedna kopia danej krotki.

- Wyjmowanie krotki z TS, nadając ewentualnie wartości niektórym z jej pól. Jest to wykonywane przez operację *in*. Jeżeli (*example"*, 4) znajduje się w TS, to *in* (*"example"*, 4) usunie ją z TS, natomiast *in* (*"example"*, ? *val*) oprócz usunięcia tej krotki z TS nada zmiennej *val* wartość 4 – argument operacji *out* został dopasowany do krotki z TS. Mówimy, że krotka *A* może być dopasowana do krotki *B* z TS wtedy i tylko wtedy, gdy obie mają tyle samo pól i dla każdej pary odpowiednich pól *A* (*i*) *B* (*i*)

Pracę wykonano w ramach grantu KBN nr: 8.0077.91.01.

[1] Linda jest zastrzeżonym znakiem handlowym firmy Scientific Computing Associates

[2] POSYBL-1.102 jest ogólnodostępną implementacją C-Lindy na sieć stacji roboczych SUN. Został napisany przez G. Schoinasa z University of Crete. POSYBL-1.102 można otrzymać w sieci Internet pod adresem *ariadne.csi.forth.gr*.

[3] Terminu tego użyto w [6]

ich typy są zgodne oraz zachodzi jedna z następujących trzech sytuacji:
1) wartości $A(i)$ i $B(i)$ są ustalone i takie same:
2) $A(i)$ ma potencjalną, a $B(i)$ – ustaloną wartość. Wartość $B(i)$ wtedy jest nadawana polu $A(i)$;
3) oba pola mają wartości potencjalne.

Jeżeli w TS jest więcej niż jedna pasująca krotka (np. wykonywana operacja to *in*(*"example"*, ? *val*), a w TS znajdują się (*"example"*, 2) i (*"example"*, 3), to zostanie wybrana niedeterministycznie jedna z nich. Jeżeli nie pasuje żadna – operacja *in* czeka aż do chwili, gdy w TS pojawi się odpowiednia, pasująca krotka.

● Odczytywanie krotki z TS – operacja *rd*. Osiąga dokładnie taki sam efekt jak *in*, ale z jedną, istotną różnicą: krotka będąca argumentem *rd* nie jest usuwana z TS, jak to odbywa się przy *in*. Podobnie jak *in*, *rd* powoduje zablokowanie wykonującego tę operację procesu aż do chwili, gdy będzie dostępna pasująca krotka.

● Utworzenie tzw. **aktywnej krotki** – operacja *eval*. W odróżnieniu od *in*, gdzie pola krotek muszą mieć ustalone wartości, pola krotki będącej argumentem *eval* mogą być wywołaniami funkcji. Funkcje te nie są obliczane przed ich włożeniem do TS, lecz po włożeniu – powoduje to powstanie nowych wątków lub, np. w systemie POSYBL-1.102, procesów wykonywanych współbieżnie. Wartość zwrócona przez funkcję stanie się wartością odpowiedniego pola krotki. Po obliczeniu wartości każdego z pól krotka taka staje się krotką pasywną, nie różni się niczym od włożonych do TS przez *out* i jest dostępna przez *in* i *rd*. Na przykład:

eval(*"first_eval"*, *sqrt*(4.0), *my_fun*(1, 2, 3))

spowoduje włożenie do TS aktywnej krotki, która po zakończeniu obliczania wartości swoich pól przeistoczy się w (*"first_eval"* 2.0, 6) (*my_fun*(1, 23) zwraca 6). Oczywiście, funkcje będące polami krotki-argumentu *eval* mogą korzystać z operacji *in*, *out*, *rd* a także *eval*, zmieniając w ten sposób dynamicznie liczbę procesów.

Oprócz wymienionych czterech podstawowych operacji, dostępne są jeszcze predykatowe wersje *in* i *re* – *inp* i *rdp*. Jeżeli pasująca krotka znajduje się w TS, *inp* i *rdp* działają tak samo jak *in* i *out*, zwracając dodatkowo wartość 1. Jeżeli w danej chwili nie można dopasować argumentu *inp* (*rdp*) do żadnej z krotek w TS, zwracana jest wartość 0, a polom o wartościach potencjalnych nie jest nadawana wartość.

Linda nie jest samodzielnym językiem programowania, lecz stanowi niezależny zbiór operacji, który możemy dodać do istniejącego już języka. Otrzymujemy w ten sposób równoległy język ze wspólnym dla wszystkich procesów dostępem do przestrzeni krotek i możliwością dynamicznego generowania nowych procesów, bez ograniczania mechanizmów kontroli obliczeń wyjściowego języka. Istnieją implementacje Lindy oparte na Fortranie, C, PostScript-cie, Moduli czy Prologu. Znając np. C wystarczy tylko zapoznanie się z działaniem wyżej opisanych operacji Lindy, aby pisać i uruchamiać współbieżne programy przystosowane do wykonania na sieci stacji roboczych.

Warto zauważyć, że elementy składające się na całość Lindowego programu są połączone w najmniejszym możliwym stopniu [1], odwrotnie niż np. w opartym na CSP Occamie. Procesy w Occamie są ściśle połączone, zarówno przestrzennie (dane muszą być przesyłane przez określony kanał komunikacyjny do określonego odbiorcy), jak i czasowo (proces wysyłający informację czeka aż zostanie ona odebrana). Te zależności nie istnieją w Lindzie. Po włożeniu krotki do TS proces nie interesuje się już jej losami – proces-odbiorca jest nieznany i może zostać uruchomiony dopiero po zakończeniu procesu-nadawcy. Linda realizuje deklaratywny sposób komunikacji międzyprocesowej – z informacją nie jest związany czas jej powstania ani adres odbiorcy. O tym, czy dany proces może pobrać krotkę z TS decyduje dopasowywanie. Wyłącznie kształt krotki, czyli liczba i zawartość pól, a nie miejsce lub czas powstania określają, czy proces otrzyma krotkę. Model Lindy – model luźno połączonych procesów – pozwala programiście myśleć o tworzonych aplikacjach na wyższym poziomie abstrakcji.

System POSYBL-1.102

W systemie POSYBL-1.102 wirtualna pamięć dzielona, jaką stanowi przestrzeń krotek, oraz operacje na niej zostały zaimplementowane przy użyciu standardowych mechanizmów komunikacji opartych na „socketach" (ang. socket), dostępnych w systemach opartych na BSD 4.2. Przestrzeń krotek jest rozproszona po sieci i obsługiwana przez UNIX-owe demony (w każdej stacji po jednym) nazywane *Tuple Managers*. Tworzenie wykonywalnych wersji programów polega na użyciu standardowego kompilatora *cc* języka C i dołączeniu jednej biblioteki. Pozwala to na wykorzystywanie w programach wszystkich dostępnych funkcji języka C. Program zawierający wywołania operacji Lindy może być wykonywany wtedy, gdy uruchomione zostały wcześniej *Tuple Managers*, ale oprócz tego nie różni się niczym od innych programów napisanych w C i tak jak one może być dołączany do innych środowisk umożliwiających programowanie hybrydowe.

Występują następujące różnice między POSYBL-1.102, a opisanym w poprzedniej części standardem Lindy:
● pierwsze pole krotki musi mieć wartość – nie może mieć wartości potencjalnej (dotyczy operacji *in* i *rd*),
● argumentem *eval* nie jest nazwa funkcji programu, lecz nazwa wykonywalnego pliku. Powoduje to, że *eval* ma, podobnie jak funkcja *exec* w języku C, kilka odmian (dokładny opis w języku dokumentacji systemu), natomiast jej argument staje się pełnoprawnym procesem UNIX-owym,
● nie są zaimplementowane *inp* i *rdp*. (Autor *doesn't find them very useful* [3]),
● pola krotki będącej parametrem *out* muszą mieć nadane wartości,
● przyjęto sposób zapisu pól krotki wymuszający na programiście podanie ich typu. Zmienne lub stałe będące polem krotki muszą być argumentem pewnej funkcji opisu typu. Dla ilustracji podamy znaczenie tylko tych funkcji opisu typu, które występują w naszych przykładach. Pola mające wartość zawsze muszą mieć postać l*type* **(value)**, gdzie *type* to *int* lub *string*. Funkcje opisujące typ pola o wartości potencjalnej, to **qlint (& int_val)** lub **qlstring (& char_ptr)**. Tak więc krotka (*„abc"*, 1, ? *int_val*) jest w systemie POSYBL-1.102 zapisywana jako (*lstring* (*„abc"*), *lint* (1), *qlint* (ξ *int-val*).

Programowanie w Lindzie

Wykorzystując znane i proste zagadnienia programowania równoległego chcielibyśmy przedstawić ich przykładowe rozwiązania przy użyciu systemu POSYBL-1.102. Zaprezentowane programy pokazują różne techniki programowania w Lindzie w sposób możliwie nieskomplikowany. Wybrane algorytmy są celowo bardzo proste, aby uwagę Czytelnika skoncentrować na wykorzystaniu Lindy. Z tych samych powodów, aby kod był jak najbardziej przejrzysty, nie ma np. kontroli poprawności argumentów.

Wzajemne wykluczanie procesów

Używając operacji *in* i *out* możemy w prosty sposób zaimplementować semafory binarne:

```
in(lstring("access_granted"));   (* proces czeka na krotkę*)
use_resource ();                 (* korzystanie z zasobu*)
out(lstring("access_granted"));  (* zwolnienie dostępu*)
```

Jeżeli początkowo w TS znajduje się jedna krotka (*"access_granted"*), to wykonywanie operacji *in* i *out* gwarantuje, że w danej chwili zasobu używa co najwyżej jeden proces. Jeżeli zasób jest zajęty, w TS nie ma krotki (*"access_granted"*) i procesy czekają na jej pojawienie się w TS.

Chcielibyśmy zwrócić uwagę na fakt, że Linda nie kolejkuje procesów zablokowanych na *in* – żądaniu pobrania krotki, której nie ma w TS. Może to prowadzić do sytuacji, w której „pechowy" proces nie uzyska nigdy dostępu do zasobu. Problem ten można rozwiązać w następujący sposób:

- proces inicjujący korzystanie z zasobu wykonuje:

```
out(lstring("access_granted") lint(0));
out(lstring("number_in_queue"), lint(0));
```

- proces zamierzający korzystać z zasobu wykonuje:

```
in(lstring("number_in_queue"), qlint(& my_number));
out(lstring("number_in_queue), lint(my_number + 1));
in(lstring("access_granted"), lint(my_number));
.......              (* korzystanie z zasobu *)
out(lstring ("access_granted"), lint(my_number + 1));
```

Proces ten zapamięta swój numer w kolejce i będzie czekał na przyznanie mu prawa dostępu do zasobu, wiedząc, że każdy proces, który pobrał krotkę (*"number_in_queue"*,...), także po tym pobraniu skorzysta z zasobu.

Rozproszenie obliczeń – schemat *Master-workers*

Załóżmy, że nasz program ma obliczać sumę *k*-tych potęg liczb naturalnych od 1 do *n*. Możemy to rozwiązać pisząc dwa programy: *MASTER* oraz *WORKER*.

Argumentami programu *MASTER* są kolejno: największa potęgowana liczba, wykładnik potęgi oraz liczba uruchamianych procesów *WORKER*. Pętla oznaczona *M*1 umieszcza w przestrzeni krotek liczby będące argumentami funkcji potęgującej. Zostaną one potem pobrane przez uruchamiane w pętli *M*2 procesy *WORKER* (*W*2). Pętla *M*3 zbiera rezultaty, dostarczane do przestrzeni krotek przez *WORKER-ów* (*W*3). Z punktu widzenia programu *MASTER* procesy wykonujące potęgowanie są nierozróżnialne – pobierając wynik z TS nie wiadomo, który *WORKER* go obliczył. Pętla *M*4, działająca już po obliczeniu szukanej sumy, generuje dla każdego procesu *WORKER* po jednej krotce (*"data"*, 0), która jest sygnałem do przerwania wykonywania pętli *W*1, a tym samym również procesu *WORKER*.

```
main(argc, argv)  /* program MASTER */
 int argc;
 char * argv[];
  {
   int n = atoi(argv[1]), workers_num = atoi(argv[3]), i, sum = 0, val;

  /*M1*/  for (i = 1; i <= n; i++) out(lstring("data"), lint(i));
  /*M2*/  for (i = 0; i < workers_num; i++)
                /* #1 to Unixowa sciezka dostepu do programu WORKER */
                eval_l("#1/worker", argv[2], (char *) 0);
  /*M3*/  for (i = 1; i <= n; i++)
            {
                in(lstring("result"), qlint(&val));
                sum += val;
            }
          printf(" suma wynosi : %d\n", sum);
  /*M4*/  for (i = 0; i < workers_num; i++) out(lstring("data"), lint(0));

    return 0;
   }

main(argc, argv) /* program WORKER */
 int argc;
 char * argv[];
  {
   int k = atoi(argv[1]), i, number, result;

   /*W1*/ do {
   /*W2*/       in(lstring("data"), qlint(&number));
                if (number)
                  {
                    result = number;
                    for (i = 1; i < k; i++) result *= number;
   /*W3*/         out(lstring("result"), lint(result));
                  }
              }
          while (number);

    return 0;
   }
```

Przedstawiony schemat *MASTER-WORKER* jest zapisywany bardzo czytelnie i prosto w Lindzie – nie trzeba np. pisać procedur zapewniających komunikację między nadzorującym obliczenia *MASTER-em* i wykonującymi je *WORKER-ami*.

Elegancja ma jednak swoją cenę – operacje Lindy na przestrzeni krotek nie pozwalają na przekazywanie informacji bezpośrednio między procesami, a każda krotka, zanim trafi od nadawcy do odbiorcy, musi znaleźć się najpierw w TS. Może to mieć negatywny wpływ na efektywność programu. Zasygnalizowany program przestaje praktycznie istnieć przy zagadnieniach związanych z rozproszonymi obliczeniami, np. fraktali czy generowania obrazów metodą *ray-tracingu*, gdzie umowny stosunek liczby przesyłanych informacji do właściwych problemowi obliczeń jest znikomo mały.

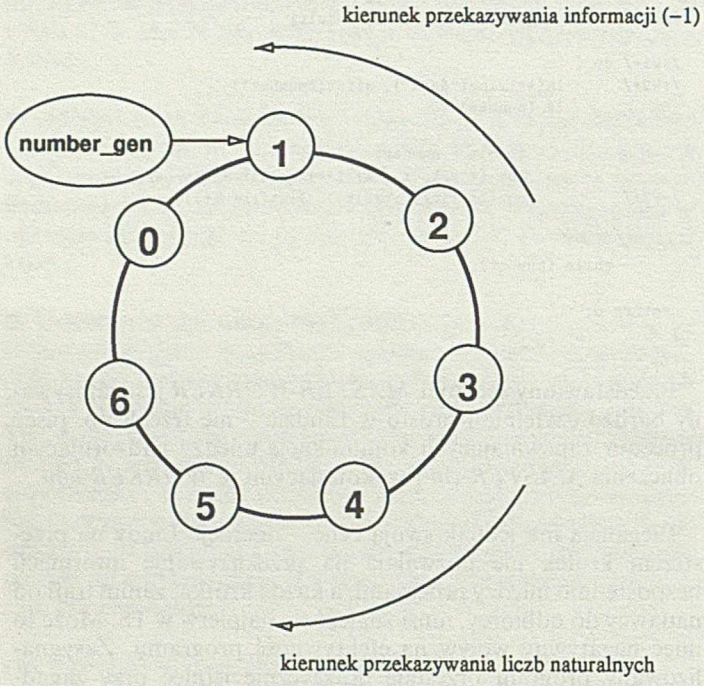
Komunikacja między procesami

W Lindzie sposobem przesyłania informacji między procesami jest umieszczenie zawierającej ją krotki w TS. Może ją odczytać jeden proces *in* albo dowolna ich liczba *rd*. Chcąc przesłać dane do konkretnego procesu, rozszerzamy przesłaną krotkę o pole jednoznacznie identyfikujące proces-odbiorcę. Możemy w ten sposób utworzyć najbardziej odpowiednią dla naszych potrzeb **wirtualną topologię** połączeń między procesami. Topologia taka może być dynamicznie zmieniana.

Interesujący przykład wykorzystania tej możliwości możemy znaleźć w [2], gdzie program obliczający liczby pierwsze metodą sita Eratostenesa tworzą procesy (krotki aktywne) połączone potokowo (ang. *pipe*). Każdy proces związany jest z określoną znalezioną przez program, liczbą pierwszą i eliminuje (odsiewa) jej wielokrotności ze strumienia liczb, generowanego przez proces wcześniejszy w łańcuchu.

Pokażemy przykład wykorzystania Lindy do zapewnienia międzyprocesowej komunikacji typu **pierścień** (ang. *ring*). Zagadnienie to realizują trzy programy, *start_ring*, *in_ring* i *number_gen*. Proces *start_ring* uruchamia generator liczb, *number_gen* (*D*1), a następnie pobiera nadsyłane do siebie informacje, wypisując odpowiednie komentarze. Krążące między procesami krotki są postaci (*process_num*, *int_data*)), gdzie *process_num* jest unikalnym numerem procesu, do którego dana krotka została wysłana. Pierwszą akcją zamierzającego włączyć się do pierścienia procesu *in_ring* jest przydzielenie sobie

numeru-identyfikatora *my_num*. Następnie jest wysyłana informacja do procesu bezpośrednio poprzedzającego w danej chwili *start_ring* (*E*2). Potem pętla *E*3 przekazuje otrzymane liczby dodatnie do procesu o numerze (*my_num* + 1) lub, o ile taki nie włączył się jeszcze do pierścienia, do *start_ring*. Pierwsza otrzymana wiadomość −1 (*E*4, *E*5), to informacja dla procesu, że kolejny proces *in_ring* zamierza włączyć się w pierścień. Wartość −1 będzie przesyłana w kierunku malejących numerów, tak jak to pokazano na rys. 1.



Rys. 1. Ustawienie procesów w pierścień

● *start_ring* – proces rozpoczynający obliczenia:

```
main()
 {
   int msg;

   out(lstring("next number"), lint(1));
   eval_l("#1/number_gen", NULL);

   do {
        in(lint(0), qlint(&msg));
        if (msg < 0) printf(" New process is welcome\n");
        else printf(" Just received : %d\n", msg);
      }
   while (msg < 1000);
```

● *in_ring* – proces dołączający do wirtualnego pierścienia. Jego następnikiem będzie proces o numerze zero:

```
main()
 {
    int my_num, next_num = 0, msg;

 /*E1*/   in(lstring("next number"), qlint(&my_num));
          out(lstring("next number"), lint(my_num + 1));
 /*E2*/   out(lint(my_num - 1), lint(-1));

 /*E3*/   do {
              in(lint(my_num), qlint(&msg));
              printf("message received : %d\n", msg);
 /*E4*/       if (msg == -1)
               {
 /*E5*/          if (!next_num) next_num = my_num + 1;
                 out(lint(my_num - 1), lint(-1));
               }
              else out(lint(next_num), lint(msg));
            }
          while (msg < 1000);
    }
```

● *number_gen* – proces uruchamiany przez *start_ring*. Jego zadaniem jest generowanie ciągu liczb naturalnych:

```
main()
 {
    int count = 1;

    while (count < 1000)

       { out(lint(1), lint(count++)); sleep(3); }
 }
```

W powyższym programie porządek nadchodzenia liczb do *start_ring* może nie mieć nic wspólnego z kolejnością generowania ich przez *number_gen*. Często takie nieuporządkowanie informacji jest niepożądane. Chcąc zapewnić pobieranie krotek w kolejności ich pojawiania się w TS, możemy posłużyć się techniką nazywaną przez twórców Lindy strumieniami (ang. *streams*) [2]. Niech elementy zbioru danych mają postać (*"stream", int_value*). Rozszerzamy te krotki do postaci (*"stream", index, int_value*), gdzie *index* ma wartość 0 dla pierwszego elementu strumienia i przyjmuje kolejne wartości naturalne dla krotek następnie wygenerowanych. Oprócz tego w TS znajdują się dwie krotki: (*"head", head_val*) i (*"tail" "tail_val*), zawierające indeksy odpowiednio ostatniej pobranej oraz pierwszej nie wygenerowanej krotki. Zauważmy, że mówiąc o informacji **pobranej** możemy mieć na myśli zarówno rzeczywiste pobranie krotki (procesy konsumujące strumień działają wolniej niż producenci), jak i jej wyłączną rezerwację przez określony proces (konsumenci wyprzedzają producentów). Możemy teraz zdefiniować:

1. operację inicjacji strumienia:
out(*lstring*("*head*"), *lint*(−1));
out(lstring("*tail*"), *lint*(0));

2. operację pobrania elementu strumienia:
in(lstring("*head*"), *qlint*(& *head_val*));
out(*lstring*("*head*"), *lint*(*head_val* + +));
in(*lstring*("*stream*"), *lint*(*head_val*), *qlint*(& *int_val*));

3. operację dopisania elementu *x* na koniec strumienia:
in(*lstring*(„*tail*"), *qlint*(& *tail_val*));
out(*lstring*(„*tail*"), *lint*(*tail_val* + 1));
out(*lstring*(„*stream*"), *lint*(*tail*), *lint*(*x*));

Bariera

Na zakończenie przeglądu wybranych metod programowania w Lindzie chcielibyśmy pokazać rozwiązanie następującego zagadnienia: istnieje pewna grupa procesów *G*1, z których każdy uruchamia losową liczbę identycznych procesów, tworzących następnie grupę *G*2. Procesy z *G*2 są zablokowane aż do momentu, gdy każdy proces z *G*1 przekaże do procesu-koordynatora obliczeń informację o liczbie procesów potomnych. Następnie grupa *G*2 staje się grupą *G*1 i, o ile liczba wchodzących w jej skład procesów jest niezerowa, postępowanie jest kontynuowane.

Jest to przykład problemu, w którym pojawia się pojęcie **bariery.** Wszystkie procesy muszą zakończyć określone działania (pokonać barierę), aby mogły przejść do następnego etapu obliczeń. Innym podobnym zagadnieniem może być prosty algorytm symulacji ruchu planet. Obliczanie położeń obiektów w danej chwili można rozpocząć dopiero po zakończeniu wyznaczania sił grawitacyjnych dla poprzedniego stanu układu. Jeszcze innym zagadnieniem, związanym z koniecznością pokonania bariery, jest przeszukiwanie grafu wszerz.

W naszym przykładzie proces zarządzający obliczeniami *coord* uruchamia początkowo jeden proces *kid_process* (*A*1),

tworzący *G*1. Wysłana informacja *enable* (*A*2, *A*6) jest dla otrzymującego (*B*1) ją procesu sygnałem o pokonaniu bariery, a więc o możliwości rozpoczęcia działania. Proces *kid_process* z pewnym prawdopodobieństwem (*B*0, *B*2) tworzy kilka swoich kopii (*B*3) i następnie zwraca informację o ich liczbie (*B*4) do *coord* (*A*5). Po zebraniu wszystkich takich informacji (*A*4), pętla *A*6 uruchamia zablokowane (*B*1) procesy tworzące nową grupę.

● koordynator obliczeń:

```
main()
  {
    int processes = 1, i, new_processes, new_ones, step = 0;

  /*A1*/  eval_l("#1/kid_process", NULL);
  /*A2*/  out(lstring("enable"));
  /*A3*/  while (processes)
            {
              printf(" step : %d, processes : %d\n", step++, processes);
              new_processes = 0;
  /*A4*/    for (i = processes; i; i--)
              {
  /*A5*/        in(lstring("new ones"), qlint(&new_ones));
                new_processes += new_ones;
              }
  /*A6*/    for (i = new_processes; i; i--) out(lstring("enable"));
              processes = new_processes;
            }
          return 0;
 }
```

● *kid_process* – proces, którego zadaniem jest uruchamianie losowej liczby swoich kopii:

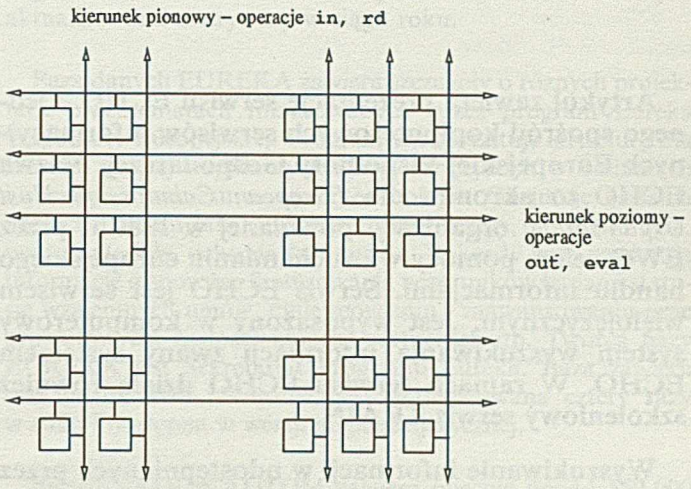
```
main()
 {
 /*B0*/    int kids_number = (getpid() * getpid()) % 10, i;

 /*B1*/    in(lstring("enable"));
 /*B2*/    if (kids_number > 4) kids_number = 0;
 /*B3*/    for (i = kids_number; i; i--) eval_l("#1/kid_process", NULL);
 /*B4*/    out(lstring("new ones"), lint(kids_number));

           return 0;
 }
```

Kierunki rozwoju Lindy

Linda nie jest systemem zamkniętym, składającym się tylko z przestrzeni krotek i niewielkiego zbioru prostych operacji na tej przestrzeni. W wielu ośrodkach są tworzone zarówno nowe, rozbudowane wersje Lindy, jak i specjalne, dedykowane temu systemowi architektury. Przykładem ciekawej implementacji Lindy może być opisana w [7] Kernel Linda, przeznaczona do programowania systemowego. Została ona zaprojektowana z myślą o wykorzystaniu jako *runtime system* dla standardowej Lindy. Procesor Lindy generuje wtedy żądania nie do biblioteki Lindy, ale do systemu Kernel Linda. Pozwala to na włączanie programów napisanych w Lindzie do systemu operacyjnego opartego na Kernel Linda.

Ważnym kierunkiem prac nad Lindą jest ujednolicanie typów pól krotek dla implementacji opartych na równych językach bazowych. Celem jest, aby na przykład [7] wygenerowana przez C++ przestrzeń krotek mogła być dostępna dla programu napisanego w Lindzie opartej na PostScript-cie. Zwiększany jest też zakres dostępnych typów pól krotek – od tak oczywistych roszczeń, jak typ struktury czy tablicy, aż do bardzo ciekawej koncepcji, aby samo pole krotki mogło być przestrzenią krotek [4]. Zaletą tego oryginalnego pomysłu jest stworzenie określonej hierarchii rozłącznych przestrzeni krotek, pozwalającej na projektowanie obliczeń o różnych stopniach modularności. Ponadto zastosowanie zwykłych Londowych operacji na krotkach o polach będących przestrzeniami krotek daje możliwość manipulowania nie tylko zwyczajnymi danymi, ale również całymi procesami obliczeń.

Opis sprzętowych prac nad wieloprocesorową Maszyną Lindy (*Linda Machine*) jest podany np. w [7]. Proponuje się ustawienie *N***N* procesorów w kwadrat o boku *N* tak, aby każdy procesor mógł przekazywać informacje w dwu prostopadłych kierunkach (rys. 2).



Rys. 2. Struktura połączeń procesorów w Maszynie Lindy

Tak powstała siatka procesorów jest zanurzona logicznie w przestrzeni krotek. Każdy procesor ma dwie części: część wykonującą obliczenia oraz koprocesor Lindy, zarządzający właściwymi Lindzie operacjami. Procesory nie komunikują się ze sobą w trybie *point-to-point*. Rozkazy *out* i *eval* powodują, że kopie krotki trafiają do lokalnych pamięci wszystkich procesorów leżących w tym samym wierszu, co procesor wykonujący daną operację. Podobnie, żądanie pobrania krotki (czyli *in* i *rd*) jest przekazywane do wszystkich procesorów z kolumny procesora czekającego na krotkę. Powoduje to znaczny wzrost efektywności systemu: jeżeli wykorzystuje się *N***N* procesorów ustawionych w opisany sposób, to w poszukiwaniu krotki należy wysłać pytanie do *N* procesorów. Alternatywą może być przekazywanie operacji *out* do wszystkich procesorów (wtedy *in* musi wykonać tylko lokalne przeszukiwanie przestrzeni krotek), przechowywanie jedynie lokalnej kopii argumentu *out* (wtedy wykonanie *in* byłoby związane z wysłaniem pytań do wszystkich procesorów) lub dowolna inna, **prostokątna** siatka procesorów. Przyjmując, że liczba operacji *out* jest równa liczbie wywołań *in*, można udowodnić, że rozwiązanie pośrednie, czyli właśnie ustawienie *N***N* procesorów w kwadrat jest najlepsze spośród wszystkich możliwych prostokątów o bokach *A* i *B* takich, że *A***B* = = *N***N*. Oprócz efektywności, powstały model Maszyny Lindy ma jeszcze jedną ważną cechę: każdy element przestrzeni krotek znajduje się jednocześnie w pamięciach lokalnych *N* procesorów.

Opisane prace, mające na celu ulepszenie i rozbudowanie Lindy oraz konstrukcję specjalnych architektur, mogą dać w rezultacie bardzo wydajne, a jednocześnie proste w użyciu narzędzie przetwarzania rozproszonego.

W pierwszym akapicie artykułu przedstawiliśmy problem podjęcia decyzji: będę programował w *X*. Mamy nadzieję, że zachęciliśmy Czytelników do podstawienia za zmienną *X* napisu Linda. Warto!

dokończenie na s. 16

DANUTA BARTOSZEK
Wyższa Szkoła Inżynierska
Radom

ECHO – komputerowy serwis informacyjny

Artykuł zawiera prezentację serwisu ECHO – jednego spośród komputerowych serwisów informacyjnych Europejskiej Wspólnoty Gospodarczej. Nazwa ECHO to akronim dla *European Commission Host Organisation*, organizacji powołanej w 1980 r. przez EWG celem pomocy w uruchamianiu europejskiego handlu informacjami. Serwis ECHO jest serwisem wielojęzycznym. Jest wyposażony w komputerowy system wyszukiwania informacji zwany systemem ECHO. W ramach serwisu ECHO działa również szkoleniowy serwis TRAIN.

Wyszukiwanie informacji w udostępnianych przez ECHO bazach danych jest obecnie bezpłatne (z wyjątkiem bazy TED). Pozwala to wykorzystywać ten system w celach szkoleniowych. Użytkownik ma w tym przypadku dostęp w trybie on-line do baz danych z dziedziny badawczej i rozwojowej działalności Wspólnoty; może on także – w bazach wspomagających pracę użytkownika – dowiedzieć się o serwisach i bazach danych z różnych dziedzin. Np. w grupie baz dotyczących gospodarki i handlu może otrzymać informacje o kontraktach handlowych, ustawodawstwie europejskim, warunkach pracy i sytuacji na rynku pracy Wspólnoty.

Zgodnie z przyjętą zasadą, użytkownikiem serwisu ECHO może być każda osoba fizyczna lub instytucja, która zawarła umowę z odpowiednią komisją Wspólnot Europejskich. Klient (pełnoprawny użytkownik) jest zobowiązany do ścisłego przestrzegania warunków umowy.

Dlaczego powstał serwis informacyjny ECHO?

Serwis informacyjny jest to – ujmując najogólniej – zespół ludzi i środków technicznych zwłaszcza komputerowych, przeznaczony do świadczenia usług z zakresu dostarczania informacji. Pomysł utworzenia jednolitego komputerowego serwisu informacyjnego dla Wspólnoty powstał w latach siedemdziesiątych. Motywacją do realizacji takiego przedsięwzięcia było [1, 2]:

- przypuszczenie, że na pozór podobne cechy systemów wyszukiwania informacji (w rzeczywistości różne już na poziomie podstawowym) powodują zamieszanie i zawodzą użytkowników;
- korzystanie przez członków Wspólnoty z różnych serwisów informacyjnych;
- zgłaszanie licznych krytycznych uwag członków Wspólnoty wykorzystujących już wyniki przeszukiwań baz;
- wyrażanie przez członków Wspólnoty chęci posiadania jednej standardowej metody konwersacji w systemach wyszukiwania informacji,
- wyrażanie chęci wykorzystywania dostępnych, światowych zasobów informacji za pomocą jednego, standardowego języka poleceń, jako uniwersalnego narzędzia wzajemnej komunikacji.

Komisja Wspólnot Europejskich CEC została upoważniona do zbadania i rozpatrzenia różnic między eksploatowanymi systemami wyszukiwania informacji. Rezultatem tych działań był język poleceń CCL (*The Common Command Language*), a efektem końcowym – m.in. płatny serwis EUROBASES i bezpłatny serwis ECHO. Prace projektowe nad CCL zakończono pod koniec 1977 r., a pierwszą implementację tego języka w 1979 r. Zgodnie z założeniami, serwis ECHO ma demonstrować możliwości i korzyści finansowe płynące z niezależnej od odległości taryfy za usługi telekomunikacyjne, a także łatwość dostępu do baz danych.

Serwis ECHO jest szczególnie pomocny w obrębie programu IMPACT (*Information Market Policy Actions*), gdzie służy do uruchamiania handlu informacjami. Według Michela Carpentier – naczelnego dyrektora Generalnej Dyrekcji (DG XIII) Komisji Telekomunikacji, Przemysłu Informacyjnego i Innowacji Wspólnot Europejskich – głównymi celami ECHO są [2]:

- dostarczanie tematycznych, wielojęzycznych informacji o usługach informacyjnych Wspólnoty (np. baza I'M GUIDE),
- poradnictwo oraz możliwość praktyki w wykorzystaniu z komputerowych serwisów informacyjnych (szczególnie objęcie obszarów, gdzie dostęp do komputerowej informacji nadal nie jest powszechny),
- pełnienie funkcji rynku zbytu dla czołowych serwisów (instytucji) informacyjnych.

Możliwości serwisu ECHO – bazy danych

Podstawowymi walorami serwisu ECHO są:

- wielojęzyczne wersje zgromadzonych informacji:
- możliwość wyszukiwania informacji w wybranym, spośród dostępnych, języku narodowym,
- możliwość bezpłatnego dostępu on-line w trybie do baz danych (obecnie – z wyjątkiem bazy TED),
- dysponowanie sprawnie działającym, nowoczesnym systemem szkolenia użytkowników,
- możliwość pomocy przy rozwiązywaniu problemów użytkownika (np. "the ECHO help desk", robot MAX).

Siedziba serwisu ECHO znajduje się w Luksemburgu, gdzie w ośrodku komputerowym serwisu jest zainstalowany system ECHO. Użytkownik może eksploatować ten system z dowolnego miejsca naszego globu za pomocą odpowiedniego oprogramowania komunikacyjnego.

System ECHO jest dostępny przez 24 godziny na dobę i przez 7 dni w tygodniu. Dzięki bezpłatnym usługom jest on głównie przeznaczony dla „nowicjuszy" w zakresie przeszukiwania baz danych w trybie bezpośrednim. Zgodnie ze swymi założeniami, serwis nie prowadzi działalności komercyjnej, jednakże w przy-

padku niektórych baz, CEC (Komisja Wspólnot Europejskich) zezwala również na komercyjne wyszukiwanie informacji. We współpracy z płatnym serwisem EUROBASES, dostęp i wykorzystanie takich baz są dokładnie kontrolowane. Jeśli okaże się, że eksploatacja takich baz przynosi zyski, to bazy te są przesuwane do serwisu EUROBASES i tam udostępniane użytkownikom na zasadach handlowych.

ECHO obejmuje ponad 20 baz danych (łącznie stanowi to kilkadziesiąt wyodrębnionych zbiorów). Liczba tych baz jest zmienna, bo nadal są dołączane nowe zbiory danych, a niektóre spośród istniejących – przesuwane do serwisu EUROBASES lub czasowo niedostępne. Informacje zgromadzone w ECHO dotyczą:

- wspomagania pracy użytkownika (np. bazy: I'M GUIDE, DUNDIS. ...; baza treningowa CCL-TRAIN albo TRAIN);
- programów badawczych i rozwojowych Wspólnoty Europejskiej (np. bazy: BIOREP, CORDIS – kilka zbiorów, DOMIS, ESPRIT, EUREKA, EURISTOTE, XIII MAGAZINE, ...);
- techniki tłumaczeń (np.: EURODICAUTOM, THESAURI);
- gospodarki (np.: TED, JUSLETTER, ELISE, MISEP).

Krótką charakterystykę dostępnych w ECHO baz danych, podano w [2], aktualny katalog tych baz można wyświetlić podczas sesji użytkownika.

Każdy użytkownik systemu ECHO może np. w bazie I'M GUIDE wyszukiwać informacje o różnych bazach danych i na podstawie takiego przeszukiwania określić dostęp do 2968 baz danych zlokalizowanych w ośrodkach komputerowych kilkuset instytucji z kilkunastu krajów.

Każdy rekord z I'M GUIDE, dotyczący danej bazy, zawiera m.in. informacje dotyczące zawartości tej bazy, jej klasyfikacji, twórcy, częstości aktualizacji, rozmiaru, rodzaju nośnika udostępniania danych oraz języka ich zapisu. Z I'M GUIDE – najnowszej wersji znanej bazy DIANE-GUIDE – można dowiedzieć się również o wielu innych produktach przemysłu informacyjnego. Niektóre rekordy w bazie dotyczą serwisów informacyjnych i zawierają m.in. adresy i numery telefonów, teleksów i telefaksów kontaktowania się, identyfikatory właściciela serwisu, nazwy używanych języków poleceń.

Na uwagę zasługuje baza DUNDIS, w której m.in. są zarejestrowane systemy, serwisy informacyjne oraz bazy danych dostępne w sieciach komputerowych ONZ. Baza ta jest aktualizowana raz w roku i stanowi komputerową wersję Katalogu Baz Danych i Systemów Informacyjnych ONZ (*The Directory of United Nations Databases and Information Systems*) o dostępie w trybie on-line.

Szczególnie przydatne mogą być bazy danych obejmujące badawczo-rozwojową działalność Wspólnoty. Wśród baz tych pokaźną grupę stanowią angielskojęzyczne bazy CORDIS.

Bazy CORDIS (*Community Research and Development Information Service*) powstały w ramach programu Value. Dostarczają one informacji na temat wszystkich programów badawczych i rozwojowych Wspólnoty, a także uzyskanych wyników badań. Zawierają dane bibliograficzne o publikacjach, dokumentach i raportach dotyczących badawczo-rozwojowej działalności Wspólnoty, a także informacje dotyczące organizacji współpracujących. Jako ciekawostkę należy uznać to, że jedna z baz CORDIS, o nazwie RTD-ACRONYMS, zawiera objaśnienia akronimów i skrótów powstających podczas prac badawczo-rozwojowych Wspólnoty. W przygotowaniu jest kilka kolejnych baz, jak np. baza RTD-PARTNERS, przeznaczona do kojarzenia odpowiednich partnerów do wykonywania zaplanowanych zadań.

Baza BIOREP opisuje biotechnologiczne przedsięwzięcia realizowane w dwunastu krajach członkowskich Wspólnoty. Bazę tę zbudowano w celu identyfikacji w tej dziedzinie kierunków badań w krajach członkowskich, pogłębienia kontaktów naukowych między badaczami oraz dostarczenia podstawy koordynacji i planowania badań. Dane do BIOREP dostarcza wiele instytucji naukowych, m.in. biblioteka Holenderskiej Królewskiej Akademii Nauk (The Royal Netherlands Academy of Arts and Sciences). Baza ta, opracowana w języku angielskim, w 1992 r. zawierała ponad 7000 rekordów i jest aktualizowana cztery razy w ciągu roku.

Baza danych EUREKA zawiera szczegóły o różnych projektach oraz tematach finansowanych przez program Eureka Wspólnoty Europejskiej. Program ten kształtuje strukturę dla szeroko pojętej wspólnoty krajów europejskich w badaniach i rozwoju z zakresu nowych technologii. Obejmuje badania z takich dziedzin, jak: transport, energia, lasery, biotechnologia, nowe materiały, ochrona środowiska. Rekordy bazy zawierają informacje o sposobie finansowania uczestników badań i statusie wzajemnych umów, umiejscowieniu i zaawansowaniu prac projektowych, zastosowaniach, partnerach itp. Twórcą bazy EUREKA jest Sekretariat Programu Eureka. Baza zawiera obecnie ok. 670 rekordów, jest aktualizowana cztery razy w roku i dostępna w wersji angielskojęzycznej.

Baza EURODICAUTOM zawiera obecnie ponad 550 000 wyrazów, wyrażeń i zwrotów z zakresu słownictwa naukowo-technicznego, objaśnianych bardzo często wielojęzycznie. Umieszczono w niej także wyjaśnienia ponad 120 000 skrótów stosowanych we wszystkich urzędowych językach Wspólnoty Europejskiej. Podczas aktualizacji baza powiększa się każdego miesiąca o ok. 2000 nowych pozycji.

TED (*Tenders Electronic Daily*) jest zbiorem ofert międzynarodowych kontraktów z ponad 80 krajów. TED jest bazą płatną, codziennie aktualizowaną. Baza ta obejmuje treść suplementu *S* do dziennika urzędowego EWG *The Official Journal of the European Communities*.

W bazie ELISE znajdują się informacje o problemach zatrudnienia i pomocy w zwalczaniu bezrobocia. Dane z bazy MISEP dotyczą dokumentów publikowanych w obrębie programu MISEP, mającego na celu wymianę informacji o bezrobociu w krajach Wspólnoty. ELISE jest francuskojęzyczna, natomiast MISEP – angielskojęzyczna.

Ważną bazą jest JUSLETTER, która dostarcza zwięzłe informacje z zakresu ustawodawstwa europejskiego ze szczególnym uwypukleniem praw człowieka. W JUSLETTER prowadzony jest także rejestr wyroków sądowych. Baza jest dostępna w wersji francuskiej (JARF) oraz w wersji angielskiej (JAFE). Aktualizowana jest co tydzień.

Jak wspomniano, serwis ECHO jest wielojęzyczny. Oznacza to, że określona informacja z bazy danych może być udostępniona użytkownikowi w co najmniej jednym języku Wspólnoty. Na przykład, treści rekordów baz I'M GUIDE oraz TED są dostępne w dziewięciu różnych wersjach językowych. Mówiący robot MAX odpowiada w jednym, wybranym spośród czterech języków (angielski, niemiecki, francuski, hiszpański). Zespół pomocy z Luksemburga (*The ECHO help desk*) składa się z osób władających różnymi językami. Użytkownik prowadzi konwersację z systemem w wybranym przez siebie, jednym z ośmiu języków zachodnioeuropejskich (angielski, francuski, niemiecki, włoski, hiszpański, holenderski, duński, portugalski).

Z klasyfikacją baz wstępnie można zapoznać się przez dokumentację użytkową systemu, np. [3]. Najnowsze dane o bazach można uzyskać podczas sesji użytkownika za pomocą opcji menu, polecenia INFO języka CCL oraz bazy I'M GUIDE. Dzięki poleceniu INFO poznaje się m.in. strukturę każdej bazy danych udostępnionej przez ECHO.

Serwis ECHO korzysta z bardzo prostej w obsłudze poczty elektronicznej.

Współpraca użytkownika z systemem najczęściej jest ograniczona do wyszukiwania informacji. Czynność ta może być realizowana podczas sesji za pomocą poleceń języka CCL. Oprócz standardowego CCL w serwisie ECHO dopuszczalne są również inne języki poleceń, które są jednak związane z określonymi bazami. W celu wyszukiwania rekordów w bazie EURODICAUTOM, można np. skorzystać z języka CCL albo z języka CEC. Przeszukiwania bazy EURISTOTE można prowadzić przy użyciu CCL albo MUNE systemu. Do wyszukiwania informacji w serwisie EUROBASES służy MENU albo języki MISTRAL i BASIS. A więc początkowy warunek używania jednego języka do komunikowania się przy korzystaniu z zasobów informacji, nie jest już rygorystycznie przestrzegany. W dalszej części artykułu zostanie przedstawiony jedynie język poleceń CCL (wersja 4.0).

W jaki sposób można uzyskać dostęp do ECHO?

W celu uzyskania możliwości bezpośredniego przeszukiwania udostępnianych przez ECHO baz danych, należy mieć:

- właściwy sprzęt,
- hasło dostępu do sieci transmisji danych,
- hasło dostępu do komputerowego systemu wyszukiwania informacji.

Każdy kto dysponuje komputerem (np. klasy PC) z modemem oraz odpowiednim oprogramowaniem, może próbować – wykorzystując np. międzynarodowe linie telefoniczne – przeszukiwać bazy danych ECHO. W skład oprogramowania komputera musi wchodzić program komunikacyjny, umożliwiający m.in. transmisję danych między komputerem użytkownika i głównym komputerem serwisu ECHO w Luksemburgu, za pośrednictwem sieci komputerowej lub zwykłej sieci telefonicznej. Wartości parametrów transmisji, wymagane przez program komunikacyjny podczas transmisji danych w określonej sieci, można odczytać z [2, 4].

Znanych jest wiele rozwiązań sprzętowo-programowych realizujących przebieg komunikacji między komputerem głównym a komputerem użytkownika ([1] s. 5–6, [2] s. 13, 33–37). Podane w [2] na s. 13 ogólne wymagania sprzętowe dla uzyskania dostępu do ECHO, to:

- każdy – kompatybilny z TTY – terminal wraz z modemem, umożliwiającym przyłączenie do ogólnie dostępnej sieci przesyłania danych z komutacją pakietów, tzn. PSDN (*Packet Switching Data Network*);
- mikrokomputer z interfejsem V24 lub RS232, wyposażony w modem, umożliwiający przyłączenie do ogólnie dostępnej sieci przesyłania danych z komutacją pakietów;
- każdy z istniejących terminali wideotekstu (*the Videotex terminal); dla niektórych z nich komunikacja z głównym komputerem jest możliwa tylko przez państwowe systemy wideotekstu;*
- synchroniczne urządzenia przystosowane do pracy w sieci komputerowej z protokołem X.25;
- bezpośrednia linia telefoniczna o szybkości transmisji danych 300 lub 1200 bitów/s.

Ogólnie dostępna sieć PSDN Luksemburga nazywa się LUXPAC, a np. Danii – DATAPAK.

Dysponując sprzętem i zgodą władz telekomunikacyjnych na korzystanie z wybranej sieci, dostęp do systemu ECHO można uzyskać wykonując następujące czynności:

- wybrać numer telefoniczny (+)352436428 (szybkość 300 bitów/s) lub (+)352420347 (szybkość 1200 bitów/s), jeżeli połączenie odbywa się przez międzynarodową sieć telefoniczną;
- wykorzystać numer (+)270448112 jako adres serwisu ECHO (tzw. ECHO NUA) w sieci z komutacją pakietów, jeżeli połączenie odbywa się przez ogólnie dostępne sieci z komutacją pakietów;
- dostosować się do wskazówek rodzimych serwisów wideotekstu w przypadku łączenia przez serwisy wideotekstu.

Dostęp do komputera w Luksemburgu można także uzyskać przez Akademickie Komputerowe Sieci Badawcze (*Academic Research Networks of RARE/COSINE*) przyłączone do IXI (*International X.25 Interconnection*). Adresem ECHO w sieci IXI jest numer (+)2043703004; szczegóły o parametrach transmisji podane są w [2] na s. 27.

Zachodnioeuropejczyk może korzystać z systemu ECHO za pośrednictwem europejskich serwisów wideotekstu, takich jak Teletel (Francja), BTX (Niemcy) lub Prestel (Wielka Brytania).

W [1] polecane do wykorzystania są także międzynarodowa sieć komputerowa z komutacją pakietów EURONET oraz DIANE (*Direct Access Information Network for Europe*).

Zalecane jest korzystanie z sieci komputerowych z komutacją pakietów (nazywanych także sieciami pakietowymi). Ponoszone przez użytkownika koszty transmisji danych są w tym przypadku co najmniej dwa razy mniejsze niż przy zwykłym połączeniu telefonicznym. Każda osoba zainteresowana taką drogą przesyłania danych powinna najpierw postarać się o upoważnienie (hasło), otrzymywane od władz telekomunikacji (administracji sieci).

```
NAUKOWA I AKADEMICKA SIEC KOMPUTEROWA
GANDALF STARMASTER ZAPRASZA DO PRACY

Jestes dolaczony do modem.0 Jesli masz problemy dzwon +/48/22/268000
Wprowadz nazwe GOSC lub GUEST i takie samo haslo aby wejsc do systemu
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Twoja nazwa?...xxxsir
Haslo?...
Usluga?...dpak
Napisz HELP dla uzyskania dodatkowych informacji
Napisz 00 ENTER aby zakonczyc prace
Napisz Ctrl P aby wrocic do modu komend

Polaczenie modem.0 z pad.0

command> c-270448112
command>
270448112
FAC: D9600
xgm: Connected

%  THIS IS ECHO; PLEASE ENTER YOUR CODE
%/
```

Wydruk 1

W warunkach polskich sprawnie i niezawodnie przekazuje informacje z/oraz do głównego komputera serwisu ECHO, sieć pakietowa NASK (Naukowa i Akademicka Sieć Komputerowa). Praktykowane były tu połączenia przez węzeł główny sieci

NASK na Uniwersytecie Warszawskim (protokół X.25, usługa DPAK) oraz przez węzeł w Kopenhadze. Pojawiający się podczas łączenia z komputerem głównym serwisu ECHO przykład tekstu ilustruje wydruk 1.

Po komunikacie „THIS IS ECHO; PLEASE ENTER YOUR CODE", system oczekuje na wprowadzenie hasła. Może to być:
- hasło prywatne;
- hasło tymczasowe;
- *„ECHO"*, które jest hasłem publicznym (tzw. *the public password*) ;
- „TRAINE", „TRAIND" lub „TRAINF", stanowiące hasło szkoleniowe.

Prawo dostępu do baz danych jakimi dysponuje ECHO uzyskuje się po akceptacji przez Komisję Wspólnot Europejskich tzw. umowy użytkownika. Po podpisaniu takiej umowy jest nadawane przyszłemu klientowi (pełnoprawnemu użytkownikowi) hasło prywatne.

Hasło tymczasowe można otrzymać z Luksemburga na okres jednego miesiąca z „the ECHO help deks", obecnie, nie daje ono jednak dostępu do baz CORDIS.

Publiczne hasło „ECHO" umożliwia eksploatację wybranych przez serwis ECHO baz danych. Takie hasło nadano m.in. w celu zachęcenia użytkownika do komputerowego wyszukiwania informacji. Obecnie, na hasło „ECHO" są udostępniane bazy CCL-TRAIN, I'M GUIDE, I'M NEWS ARCHIVE, DUNDIS, EUREKA oraz XIII MAGAZINE, przy czym stan ten jest zmienny. Czasami możliwe jest przeszukiwanie jedynie zawężonej liczby rekordów każdej z udostępnionych baz.

W ramach serwisu ECHO działa specjalny serwis o nazwie TRAIN. Jak już wspomniano, serwis umożliwia bezpłatne szkolenie (trening) dla początkujących w dziedzinie komputerowego wyszukiwania w trybie bezpośrednim (on line) informacji bibliograficznej. TRAIN pozwala także na używanie większości poleceń języka CCL (*Common Command Language*). Osoba początkująca powinna mieć broszurę [1], dostarczoną przez serwis. Korzystanie z TRAIN jest możliwe po rozpoczęciu sesji jednym z następujących haseł: TRAINE, TRAIND lub TRAINF. Sesja użytkownika przebiega w języku angielskim, jeżeli pracuje on na haśle TRAINE. Jeśli wprowadzono hasło TRAIND, to sesja jest niemieckojęzyczna. Po haśle TRAINF usługi szkoleniowe są w języku francuskim.

Opłaty za korzystanie z płatnych baz są zmienne. Średni koszt jednogodzinnego przeszukiwania płatnej bazy wynosi 50–100 $ (bez opłat za zajmowanie łączy telekomunikacyjnych). Średni czas przeszukiwania bazy wynosi 10–15 min.

W jaki sposób można zapoznać się z ECHO?

Z serwisem ECHO można zapoznać się m.in. przez:

- dokumentację systemu ECHO;
- czasopisma i komunikaty (np. „ECHO FACTS FOR USERS");
- seminaria i konferencje;
- bezpośrednią eksploatację systemu ECHO, tzn. polecenie INFO języka CCL (możliwe do wykonania podczas sesji na każde z haseł),
- kontakt z zespołem pomocy z Luksemburga („the ECHO help desk"),
- kontakt telefoniczny z mówiącym robotem MAX.

Podstawowa dokumentacja systemu składa się z kilkunastu broszur, np. [1], ..., [4], przy czym w [4] znajduje się opis języka CCL.

Wprowadzenie do systemu ECHO, wymagania sprzętowe, telefoniczne numery połączeń oraz wzmianki o bazach są podane w [2]. Dokumentacja użytkowa baz danych obejmuje także tzw. arkusze techniczne. Arkusz techniczny („ECHO Technical Sheet"), zawiera informacje o danej bazie i jest niezbędny do opracowywania sekwencji poleceń umożliwiających realizację przeszukiwań. Arkusz opisuje strukturę bazy, określa jej twórcę i częstość aktualizacji, podaje wersje językowe bazy i hasła dostępu, a także języki poleceń, jakich można używać. Zawiera również przykładowy wydruk rekordu oraz wzorcowy układ poleceń dla przeszukiwania bazy.

Użytkownik ECHO otrzymuje publikację „ECHO FACTS FOR USERS", z której może dowiedzieć się o nowych bazach danych, działalności personelu serwisu, konferencjach bądź wystawach. Zapowiedzi konferencji oraz wystaw można uzyskać także w zespole pomocy („the ECHO help desk") oraz podczas sesji użytkownika po zastosowaniu np. INFO EVENTS.

W [2] na s. 38 są zapisane numery połączeń telefonicznych z różnych krajów Wspólnoty do „the ECHO help desk". Ten zespół serwisu jest wielojęzyczny.

MAX jest robotem systemu ECHO próbującym interpretować głos ludzi i rozmawiać. Jego głos brzmi nieco sztucznie i rozumie niewiele słów. MAX słucha tylko wtedy, gdy nie mówi, a więc nie należy przerywać jego wypowiedzi. Od MAX-a można uzyskać informacje na następujące tematy: kurs wymiany ECU, podstawowe europejskie dane statystyczne, najnowsze wiadomości ECHO, kalendarz ECHO, programy Wspólnoty Europejskiej, ogólne pojęcia technologii informacyjnej, instytucje Wspólnoty Europejskiej, biura prasowe Wspólnoty. MAX jest przygotowany do rozmów w języku angielskim albo niemieckim. Od niedawna MAX mówi także po hiszpańsku oraz po francusku.

Charakterystyka wersji 4.0 języka poleceń CCL

Głównym celem osoby korzystającej z systemu ECHO jest wyszukiwanie z zawartości baz danych zestawu potrzebnych informacji. Osoba ta na początku ma jedynie szczątkowe (hasłowe) dane z zakresu interesującego ją zagadnienia. Poprawne umieszczenie takich danych w poleceniach języka CCL pozwoli na ich poszukiwanie w odpowiednich bazach i pozyskiwanie rekordów, zawierających potrzebne, uzupełniające informacje.

Do pracy w języku CCL przystępuje się po wybraniu z głównego menu systemu opcji języka CCL „CCL MODE". System ECHO zgłasza swą gotowość przyjęcia polecenia języka CCL przez przesłanie do użytkownika znaku ?. Po takim zgłoszeniu należy wprowadzić polecenie, które należy zakończyć wciśnięciem klawisza ENTER.

Ogólna struktura polecenia to:

<nazwa polecenia> [wyrażenie kwalifikujące]

albo

<skrót nazwy polecenia> [wyrażenie kwalifikujące]

Spis nazw oraz skrótów ważniejszych poleceń wersji 4.0 języka CCL podano w tabeli,

Nazwy i skróty ważniejszych poleceń języka CCL wersja 4.0

Nazwa	Skrót	Przeznaczenie
* BACK	B	pokazanie "wcześniejszej" części listy wyników
BASE	BAS	wybieranie i otwieranie bazy do przeszukiwania
CONTINUE	C	oczekiwanie systemu, z podpowiedzią, na odpowiedź użytkownika
* DEFINE	DEF	definiowanie standardowych własności systemu
DELETE	DEL	kasowanie profili przeszukiwania; kasowanie korespondencji z poczty elektronicznej
DISPLAY	D	wyświetlanie elementów zbioru indeksowego
FIND	F	przeszukiwanie rekordów otwartej bazy danych
HELP	H ?	prośba o pomoc (np. informacja systemowa o CCL)
INFO	INFO	m.in. uzyskiwanie ogólnych informacji o bazach
* MAIL	MAIL	korzystanie z poczty elektronicznej ECHO
MORE	M	oczekiwanie systemu, z podpowiedzią, na odpowiedź dotyczącą pokazywania dalszych wyników
* PRINT	P	drukowanie wyników w ośrodku głównym
* SAVE	SAVE	zapamiętywanie poleceń w pamięci komputera gł.
SHOW	S	pokazywanie rekordów na ekranie użytkownika
STOP	STOP	zakończenie konwersacji z systemem ("wyjście")

(Uwaga: Polecenia oznaczone przez * nie są dostępne w serwisie TRAIN).

CCL ma m.in. następujące symbole:

AND OR NOT – działań logicznych,
= > < > = < = – relacji arytmetycznych.
$ – maskowania oraz obcinania,
, . $
. . $$
. . . $$$ – działań „sąsiedztwa słów" (ang. *adjacency*),
; , – separatorów między opcjami wyrażenia kwalifikującego niektórych poleceń (np. SHOW),
TO – zakresu.

Wyrażenie kwalifikujące jest budowane przez użytkownika. Może ono składać się ze słów, a czasem być np. zapisem działania lub działań na słowach lub terminach językowych. Do ustalenia kolejności wykonywania działań należy w nim stosować nawiasy okrągłe.

Język CCL identyfikuje pola bazy za pomocą etykiet pól, a języki narodowe za pomocą kodów językowych ([1] s. 19).

Szczegółowy opis wszystkich poleceń oraz działań CCL można znaleźć w [4]. Dla przykładu, polecenie BASE:
- wybiera i otwiera bazę danych w celu przeszukiwania;
- wyświetla informacje o otwartej bazie danych;
- wyświetla listę dostępnych baz danych.

Postacie polecenia **BASE**, to:
BASE (dostarcza listę dostępnych baz danych),
BASE <nazwa bazy> [;<**uzupełnienie**>...]
(wybiera i otwiera bazę danych),
BASE <4-znakowy identyfikator bazy tzw. poolkey>
(wybiera i otwiera bazę danych).
Polecenie **BASE ?** identyfikuje otwartą bazę danych.
Uzupełnieniem może być każde z następujących trzech żądań:
ED = mm.rr TO mm.rr (ogranicza przeszukiwanie rekordów do dodanych podczas wskazanego okresu czasu, gdzie mm – miesiąc, rr – rok);
NOFIND (baza i zbiór indeksowy mogą być badane, ale przeszukiwanie nie będzie wykonane);
TL = <kod języka> (wybiera jedną wersję językową dla bazy na podstawie czteroznakowego kodu języka).

W praktyce, spotyka się np. polecenia:

BASE TED; TL=GERM
BAS EURISTOTE; ED=01.79 TO 12.80; TL=FREN.

W zapisie polecenia, między nazwą (skrótem) polecenia a wyrażeniem kwalifikującym, musi znajdować się spacja

Jak korzystać z wersji 4.0 języka CCL?

Wyszukiwanie informacji w bazie danych polega na identyfikacji tych rekordów bazy, których wybrane pola spełniają narzucone przez użytkownika warunki. Proces wyszukiwania (przeszukiwania bazy danych) w systemie ECHO następuje na skutek działania odpowiedniej sekwencji poleceń języka CCL. Identyfikacja rekordów zachodzi na podstawie zbiorów indeksowych. Każdej bazie, udostępnianej przez ECHO, przyporządkowanych jest kilka (czasem kilkanaście) zbiorów indeksowych.

Zbiór indeksowy jest zbiorem list zawierających symbole odpowiadające danym z rekordów bazy i wspomagających odszukiwanie tych danych za pomocą umieszczonych w zbiorze adresów. Budowany jest na podstawie zawartości bazy i dzięki swej strukturze służy do usprawnienia i przyśpieszenia procesu wyszukiwania informacji. Istnieją różne sposoby organizacji tych zbiorów, np. przy tworzeniu zbiorów indeksowych dla bazy bibliograficznej, często stosuje się metodę indeksowania odwrotnego (ang *inverted indexes* lub *inverted files*).

Dla pól bazy o ustalonej etykiecie można zbudować odpowiedni zbiór indeksowy. Oprócz tego, dla grupy pól bazy o określonych etykietach jest tworzony bardzo często jeden wspólny zbiór indeksów. Informacje o tym, które pola są indeksowane łącznie, można odczytać z arkusza technicznego bazy, co jest przydatne podczas redagowania poleceń dla przeszukiwania danej bazy.

Dla realizacji wyszukiwania w otwartej bazie danych, wystarczy znać następujące polecenia:

(1) **DISPLAY** <wyrażenie kwalifikujące>
gdzie wyrażenie kwalifikujące jest ciągiem znakowym, którego początkowymi elementami mogą być (opcjonalnie) etykieta pola oraz znak "=", a kolejnymi elementami – znaki terminu językowego; elementami wyrażenia kwalifikującego mogą być też: znak „$" jako symbol działania „truncation" albo „masking", symbol zakresu **TO**;

(2) **FIND** [<wyrażenie>] [<symbol działania>[<wyrażenie>]...] gdzie wyrażenie jest ciągiem znakowym, którego elementami są znaki terminu językowego albo numer wyznaczonego zbioru (znaki terminu językowego opcjonalnie poprzedzają znak „/" i listę etykiet pól, znaki terminu językowego mogą być (opcjonalnie) poprzedzone przez etykietę pola i znak „="); symbol działania, to m.in.: = , > , < , $, .,,, NOT, AND, OR;

(3) **SHOW** <opcja 1> [; <opcja 2>...]
(3') **SHOW**

Opcje polecenia SHOW zostaną opisane w dalszej części artykułu. DISPLAY wyświetla elementy zbioru indeksowego. FIND znajduje poszukiwane rekordy, a SHOW je wyświetla.

Wyrażenie kwalifikujące w FIND przypomina nieraz którąś ze znanych relacji arytmetycznych. Czasem kojarzyć się może z wyrażeniem logicznym, którego argumentami są słowa albo identyfikatory liczbowe. Zdarza się, że po nazwie polecenia wystarczy napisać słowo albo zwrot.

Następujące przykłady wyjaśniają wykorzystanie poleceń (1) – (3) oraz sposób budowania wyrażeń kwalifikujących.

Przykład 1. W angielskojęzycznej wersji bazy I'M GUIDE (poolkey – IM92) należy wyszukać liczbę rekordów, w których:
a) w polach NA (nazwa) oraz AB (streszczenie) występuje słowo ECHO;
b) w polu NA (nazwa) występuje słowo ECHO;
c) w polu LA (język) występuje kod języka francuskiego – FREN.

Przygotowanie polecenia i wyniki przeszukiwania dla a):

BAS IM92
FIND ECHO/(NA; AB)
1.00 NUMBER OF HITS IS 6

(Uwaga: HITS, to liczba mnoga od HIT (trafienie), a więc wynik przeszukiwania wynosi 6).

Przygotowanie polecenia i wyniki przeszukiwania dla b):

BAS IM92
FIND ECHO/NA
2.00 NUMBER OF HITS IS 4

Przygotowanie polecenia i wyniki przeszukiwania dla c):

BAS IM92
FIND LA = FREN
3.00 NUMBER OF HITS IS 730

Polecenia FIND w przedstawionych przykładach istotnie różnią się wyrażeniami kwalifikującymi. Taki układ wynika bezpośrednio z reguł indeksowania pól bazy. Jak wiadomo, niektórym odpowiednio dobranym grupom pól przyporządkowany jest jeden zbiór indeksowy. W przypadku bazy I'M GUIDE są to obecnie pola o następujących etykietach: NA (nazwa), AB (streszczenie), ADDR (adres), CTBR (słowa kluczowe), COMP (nazwa przedsiębiorstwa lub instytucji). Takie pola są określone wspólną nazwą *freetext* i w arkuszu technicznym bazy mają wspólne oznaczenie – **FT.**

Wiadomo, że podczas realizacji polecenia FIND są przeszukiwane odpowiednie zbiory indeksowe, z których są odczytywane numery rzeczywiste rekordów otwartej bazy (numery te są umieszczane w tzw. zbiorze numerów).

Użytkownik musi dokładnie wskazać, które ze zbiorów indeksowych powinny być przeszukiwane. Takie działanie jest w języku CCL możliwe przez zastosowanie w wyrażeniu kwalifikującym **przedrostka** (ang. *prefix*) albo **przyrostka** (ang. *suffix*).

Obowiązują tu następujące reguły:
r1) jeśli przeszukiwanie ma dotyczyć jednego pola spoza pól typu *freetext*, to w elementach ciągu znakowego – bezpośrednio przed poszukiwanym układem znakowym (np. słowami) – należy umieścić etykietę pola i znak „="; etykieta pola i znak „=" stanowią przedrostek;

r2) jeśli przeszukiwanie ma dotyczyć jednego spośród pól typu *freetext*, to w elementach ciągu znakowego – bezpośrednio po poszukiwanym układzie znakowym – należy umieścić znak „/", a po nim – etykietę pola; znak „/" i etykieta pola stanowią przyrostek;

r2') jeśli przeszukiwanie ma dotyczyć kilku spośród pól typu *freetext*, to w elementach ciągu znakowego – bezpośredniego po poszukiwanym układzie znakowym – należy umieścić znak „/" oraz listę etykiet pól; znak „/" i lista etykiet pól stanowią przyrostek;

r3) jeśli przeszukiwanie ma dotyczyć pól typu *freetext*, to nie ma potrzeby stosowania przedrostków oraz przyrostków.

Przykłady:
CT = RECREATION, AGRONOMY/AB, BSS/(NA;AB), PRESSURE.

Po rozpoznaniu istnienia przyrostka albo przedrostka, system potrafi zidentyfikować wymagane zbiory indeksowe. Często zdarza się, że dokładna postać hasła wyszukiwania (np. pełna nazwa instytucji) nie jest znana użytkownikowi. Nieraz również podane przez użytkownika wyrażenie słowne (jedno), może być zapisane w rekordach bazy na kilka różnych sposobów (uwzględniając nawet błędnie wprowadzone dane) jak np. CHOPIN F, FRYDERYK CHOPIN. Aby wyczerpująco wyszukać informacje w otwartej bazie danych, należy w takim przypadku najpierw obejrzeć podobne zapisy w zbiorze indeksowym. Do tego celu służy polecenie DISPLAY języka CCL.

DISPLAY wyświetla w porządku alfabetycznym odpowiednio numerowaną listę elementów zbioru indeksowego, rozpoczynając od żądanego w wyrażeniu kwalifikującym. W przypadku braku żądanego zapisu, lista zaczyna się elementem najbardziej podobnym w zapisie. Numer pozycji listy jest postaci „nn.nn". Dla ilustracji:

DISPLAY CT = NUCLEAR
3.01 15104 Nuclear and Radiochemistry
3.02 2805 Nuclear Explosions
3.03 12806 Nuclear Heat for chemical processes.

Symbol CT oznacza etykietę pola „słowa kluczowe". Wiele baz, dostępnych w systemie ECHO, ma pole o takiej etykiecie. Wybrany właściwy numer listy można następnie użyć w poleceniu FIND (np. FIND 3.01). W niektórych przypadkach celowe jest użycie przed poleceniem FIND polecenia DISPLAY.

Podczas budowania wyrażenia kwalifikującego dla polecenia DISPLAY obowiązują reguły r1, r3. Na przykład:

DISPLAY E. WEDEL
DISPLAY AU-MOZART W.A.

Przypuśćmy, że w wybranej, otwartej bazie danych użytkownik chce wyszukać informacje na temat produkcji fabryki E. Wedel. Z powodu niepewności zapisu nazwy firmy lub wątpliwości jej istnienia w rekordach bazy, wskazane jest wykonanie najpierw polecenia

DISPLAY $WEDEL.

Wynikiem realizacji tego polecenia jest numerowana lista elementów zbioru indeksowego, w których dane z rekordów bazy kończą się na Wedel (dopuszczalna jest lista pusta). Zastosowano bardzo pożyteczną operację obcinania (ang. *truncation*). Wynikiem działania $Wedel jest zbiór danych kończących się na Wedel.

Znak $ jest w języku CCL m.in. symbolem następujących działań:
- obcinania lewostronnego – np. $Wedel;
- obcinania prawostronnego – np. laser$.
- obcinania obustronnego – np. PZL;
- maskowania (ang. *masking* – np. la$er).

Stosowanie obcinania w wyrażeniu kwalifikującym polecenia jest wskazane np. wtedy, gdy dane słowo szukanego zwrotu może występować w liczbie mnogiej lub pojedynczej.

Jeżeli wyszukiwanie informacji dotyczy zagadnienia, dla którego słowa hasłowego opisu mają synonimy lub słowa bliskoznaczne, to podczas opracowywania poleceń CCL należy fakt ten uwzględnić przez wykorzystanie operacji logicznych.

Na przykład:

FIND HOME OR HAUSE
FIND DOM OR BUDYNEK

Operatory logiczne NOT, AND, OR są bardzo mocnym narzędziem języka wyszukiwania informacji. Można to zaprezentować następująco:

FIND NUCLEAR OR ATOMI$
1.00 NUMBER OF HITS IS 272
F NUCLEAR **AND** WASTE
2.00 NUMBER OF HITS IS 20
F NUCLEAR **NOT** WASTE
3.00 NUMBER OF HITS IS 212
F 2 **AND** ATOMIC
4.00 NUMBER OF HITS IS 15

(Uwaga: Każdy z użytych symboli AND, NOT, OR musi być obustronnie otoczony spacją).

Zdarza się, że porządek szukanego dwusłownego terminu językowego nie jest istotny (przypuszcza się nawet, że w polach rekordów jest zmienny). W takim przypadku należy to uwzględnić w wyrażeniu kwalifikującym polecenia FIND przez działanie *adjacency*, czyli użycie jako symbolu działania – znaku przecinka między kolejnymi słowami. Na przykład:

F ziemia , polska

spowoduje poszukiwanie terminów ziemia polska oraz polska ziemia.

Każde z następujących poleceń:

F Kaczorowski **AND** president

F president , Kaczorowski

F president , Kaczorowski

może doprowadzić do trzech różnych wyników przeszukiwania.

Nieraz przewiduje się, że słowa poszukiwanego wyrażenia językowego mogą być w polach rekordów bazy rozdzielone dowolnym słowem (ciągiem znakowym). W takiej sytuacji wskazane jest wykonanie kolejnych działań typu *adjacency*. Należy zatem wykorzystać symbol (operator) „$" albo ".". Na przykład:

F Kaczorowski $ president
F Kaczorowski . president.

Symbol „$" (nie mylić z *truncation*) oznacza tu działanie, którego wynikiem jest zbiór trójczłonowych (np. trójsłownych) terminów językowych, gdzie argumenty działania znajdują się w odległości jednego nieokreślonego członu (np. słowa). Symbol "." oznacza działanie, wynikiem którego jest zbiór co najwyżej trójczłonowych (np. trójsłownych) terminów językowych, gdzie argumenty działania znajdują się w odległości co najwyżej jednego nieokreślonego członu (np. słowa). Według podobnych zasad definiuje się pozostałe działania typu *adjacency*. Np. wynikiem działania "$ $" jest zbiór czteroczłonowych (np. czterosłownych) terminów językowych, gdzie argumenty działania znajdują się w odległości dwóch nieokreślonych członów (np. słów).

Wszystkie przedstawione symbole działań typu *adjacency* są, w zapisie działania, otoczone obustronnie spacją.

System wykorzystuje polecenie MORE podczas wprowadzania wielostronicowych wyników realizacji poleceń FIND oraz DISPLAY; po podpowiedzi MORE system oczekuje na odpowiedź.

Do wyprowadzenia zawartości rekordów służy SHOW. Opcje polecenia SHOW to:

S = ⟨numer zbioru wyników⟩
R = ⟨numer rekordu⟩
R = ⟨zakres numerów rekordów⟩
F = ⟨numer formatu⟩
F = ⟨etykieta pola⟩[;⟨etykieta pola⟩...]

Na przykład:

SHOW S=5;R=1 TO 5;F=AU;NA;AB

pokaże pola AU,NA,AB rekordów 1–5 ze zbioru wyników numer 5.
Polecenie (3') wyświetli w formacie standardowym wynik ostatnio zrealizowanego polecenia FIND.

W trakcie sesji użytkownik może skorzystać z polecenia HELP (skróty: H, ?), które spełnia rolę poradnika o języku CCL. Postacie tego polecenia to:

```
base IM92
BASE COMMAND ACCEPTED FOR IM92;IM GUIDE;ED=09.04.92 TO 09.04.92;
TL=ENGL
*****************************************
*        I'M  GUIDE                     *
*****************************************
  Vous pouvez commencer votre recherche en LCC maintenant, mais
si vous desirez plus d'information sur I'M GUIDE, veuillez taper:
INFO IM92
?
define tl=FREN
?
find ORTY=$ and EUROBASES
1.00  NUMBER OF HITS IS  1
?
show
1.00/000001 ECHO: -IM GUIDE /COPYRIGHT ECHO
NA    : (EUROBASES) Eurobases
ORTY  : HOST        ... Serveur de bases de donnees
        PUBL        ... Editeur
ADDR  : Commission of the European Communities
         rue de la Loi, 200
        B-1049 Bruxelles
CY    : BE          ... Belgique
TEL   : +32-2-2350001;-2350003
TLX   : COMEU B 21877
FAX   : +32-2-2360624
CP    : Mr.Glaeser
RL    : menu; mistral; basis
NUA   : 270429200; 270429121
DB    : (CELEX) Celex
DB    : (ECLAS) European Commission's Library Automated System
DB    : (INFO 92) Info 92
DB    : (RAPID) Rapid
DB    : (SCAD) Systeme Communautaire d'Acces a la Documentation
DB    : (SESAME) Sesame
***END OF SHOW***
?
stop
QUERY : 00007390/0001/        /IM92    DATE : 30.04.92   TIME :
16:31:25
%  EXC0419 1631 92-04-30 8782
%  EXC0421 8.7438
```

Wydruk 2

HELP – poradnictwo dotyczące ostatnio wykonywanego polecenia;
HELP nazwa polecenia – informacja o zastosowaniu określonego polecenia.

Oprócz tego można używać:
⟨nazwa polecenia⟩? – informacje o wykorzystaniu polecenia w bieżącej sesji;
HELP ? albo ?? – ogólne poradnictwo.

Użytkownik kończy konwersację z systemem za pomocą polecenia **STOP**. System odpowiada końcowymi rozliczeniami.

Dwa kolejne przykłady ilustrują zastosowanie języka CCL do wyszukiwania informacji.

Przykład 2. We francuskojęzycznej wersji systemu ECHO wyszukać informacje o serwisie EUROBASES. Przeszukiwanie ograniczyć do bazy I'M GUIDE. Wyniki podać w języku francuskim. Przygotowane polecenia (zapis wytłuszczonym drukiem) oraz otrzymane wyniki ilustruje wydruk 2.

Wyjaśnienie użytych etykiet pól bazy I'M GUIDE: NA – nazwa (serwisu), ORTY – typ organizacji, ADDR – adres, CY – nazwa kraju, TEL – numer telefonu, TLX – numer teleksu, FAY – numer faksu, CP – nazwisko osoby dla kontaktu, RL – nazwa języka wyszukiwania, NUA – numer NUA, DB – nazwa bazy danych.

Przykład 3. We francuskojęzycznej wersji systemu ECHO wyszukać informacje o wersji francuskiej on-line bazy danych CELEX. Przeszukiwanie ograniczyć do bazy I'M GUIDE. Wyniki podać w języku francuskim.

Przygotowane polecenia (zapis wytłuszczonym drukiem) oraz otrzymane wyniki ilustruje wydruk 3.

Wyjaśnienie użytych etykiet pól bazy I'M GUIDE: NA – nazwa bazy, CT – słowa kluczowe, AB – streszczenie, PRTY – typ bazy, DBST – rok założenia bazy, REC – liczba rekordów w bazie, RECY – liczba rekordów dołączonych w ciągu roku, UPD – częstość aktualizacji, LA – język, DBPR – twórca bazy, ME – nośnik danych, Host – właściciel, DB – nazwa bazy danych.

* * *

W kolejnym etapie pozyskiwania informacji o EUROBASES należy zdecydować się na ich wyszukiwanie o bazach ECLAS, INFO 92, RAPID, SCAD, SESAME. Do realizacji takiego celu zredagowanie poleceń jest już czynnością bardzo prostą dla każdego, kto zapoznał się z prezentowanym materiałem w serwisie ECHO.

dokończenie na s. 21

```
?
base IM92
BASE COMMAND ACCEPTED FOR IM92;IM GUIDE;ED=17.09.92 TO 17.09.92;
TL=ENGL
*****************************************
*          I'M   G U I D E              *
*****************************************
  Vous pouvez commencer votre recherche en LCC maintenant, mais
si vous desirez plus d'information sur I'M GUIDE, veuillez taper:
INFO IM92
?
define tl=FREN
?
display $celex

   6.01        6 CELEX
****END OF DISPLAY****
?
display PRTY=$
   6.01     2095 BIB        ... Bibliographique
   6.02     1899 FAC        ... Factuel
   6.03     1159 FUL        ... Texte integral
   6.04      264 GRA        ... Graphique
   6.05      518 IMA        ... Image
   6.06      157 OTPR       ... Producteur d'autres media
   6.07      439 REF        ... Reference
   6.08      219 SOU        ... Sonores
****END OF DISPLAY****
?
find PRTY=$ and CELEX/NA and ME=ON
6.00  NUMBER OF HITS IS  1
?
show
6.00/000001 ECHO:  -IM GUIDE /COPYRIGHT ECHO
NA    : (CELEX) Communitatis Europae Lex
CT    : DROIT; COMMUNAUTE EUROPEENNE
AB    : Contient le texte des traites et amendements, le droit
        communautaire derive et complementaire, les actes
        preparatoires du PE, de la CCE et du CES, les questions
        parlementaires, tous les arrets de la Cour de justice.
PRTY  : BIB        ... Bibliographique
        FUL        ... Texte integral
        FAC        ... Factuel
```

Wydruk 3

```
DBST  : 1951
REC   : 127424
RECY  : 5000
UPD   : D          ... Quotidien
PRT   : Repertoire de la legislation communautaire en vigueur
LA    : DANH; GERM; ENGL; SPAN; FREN; GREK; ITAL; DUTH; PORT
DBPR  : (CEC DG9 I BD SERVICE CELEX) Commission of the European
        Communities
MORE
        (RCC-IVEV) RCC-IVEV informatiediensten
ME    : ON         ... Base de donnees en ligne
Host  : EUROBASES, GATEWAY: EDICLINE

DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: ICEX
DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: TBD TIME SHARING
DB    : (CELEX) Celex
Host  : EUROBASES
DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: PROFILE INFORMATION
DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: CERVED
DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: CED
DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: EUROPEENNE DE DONNEES
DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: JURIS
DB    : (CELEX) Celex
MORE

Host  : EUROBASES
DB    : (CELEX) Celex
Host  : EUROBASES, GATEWAY: DATA STAR
DB    : (CELEX) Celex
***END OF SHOW***
?
stop
QUERY : 00007390/0001/        /IM92   DATE : 25.09.92   TIME :
16:33:26
%  EXC0419 1633 92-09-25 3087
%  EXC0421 22.6492
```

LESZEK KOTZIAN
Instytut Informatyki Teoretycznej i Stosowanej
Polska Akademia Nauk
Gliwice

Prosta integracja środowiska sieci NetWare ze światem UNIX

Problem łączenia komputerów pracujących z różnymi systemami operacyjnymi i sieciowymi, a używających do komunikacji różnych protokołów, doczekał się wielu rozwiązań. Klasycznym przykładem jest próba zapewnienia dostępu do zasobów maszyny UNIX-owej mikrokomputerom klasy IBM PC pracujących w sieci NetWare. Te bez wątpienia najbardziej rozpowszechnione systemy można integrować na wiele sposobów. Jednym z prostych rozwiązań jest zastosowanie produktu programowego LAN WorkPlace for DOS.

Ogólnie rzecz ujmując, idea sieci NetWare polega na udostępnieniu kilku komputerom klasy IBM PC pamięci tego samego sztywnego dysku(ów). Wydzielony komputer – serwer – obsługuje ten dysk(-i) i pełni tym samym rolę składnicy plików, które udostępnia pozostałym przyłączonym do niego komputerom – tzw. stacjom roboczym współpracującym z systemem DOS. Medium łączącym serwer ze stacjami roboczymi jest zwykle cienki Ethernet, a używane protokoły wymiany informacji to IPX/SPX.

UNIX jest systemem operacyjnym, który pozwala współdzielić nie tylko pamięć, ale również inne zasoby jednego komputera, w tym jego procesor. Pozwala to na pracę na tym samym komputerze wielu użytkowników, którzy komunikują się z nim przez dołączone terminale. Rolę terminali mogą pełnić komputery klasy IBM PC, które w zależności od „inteligencji" oprogramowania, w większym lub mniejszym stopniu wykorzystują także swoje własne zasoby. Komputery z systemem UNIX oraz pełniące rolę terminali komputery IBM PC można również wygodnie łączyć przy zastosowaniu standardu Etherneta, przy czym wymiana informacji jest oparta w tym przypadku na standardowych protokołach UNIX-a – TCP/IP.

Oba światy protokołów IPX/SPX oraz TCP/IP ze sobą nie kolidują, umożliwiając ich współistnienie na jednym medium Etherneta. Oznacza to, że do tego samego kabla można przyłączyć zarówno IBM PC, komunikujący się z serwerem NetWare, jak i IBM PC, komunikujący się z komputerami pracującymi w systemie UNIX. Jednakże te dwa światy, współdzieląc do komunikacji to samo medium, wzajemnie nie widzą się.

Integracja NetWare – UNIX

W zasadzie z każdego IBM PC można przyłączyć się albo do serwera NetWare, albo do maszyny UNIX-owej. Jednakże każdorazowa zmiana środowiska pomiędzy NetWare i UNIX wymaga restartu IBM PC, podczas gdy jednoczesny dostęp do zasobów NetWare i maszyn UNIX mógłby znacznie poszerzyć zakres ich zastosowań. Proste rozwiązanie tego problemu nasuwa się samo. Bez ingerencji w serwer NetWare lub w maszynę UNIX-ową, zmierzającej np. do ujednolicenia protokołu komunikacyjnego, wystarczyłoby wyposażyć IBM PC w oprogramowanie (nakładkę systemową), które w zależności od potrzeb, wymieniałoby informacje z konkretną maszyną, przełączając protokoł ich nadawania z IPX/SPX na TCP/IP lub odwrotnie. Naprzeciw tym potrzebom wychodzi oferowany przez firmę Novell pakiet programowy LAN WorkPlace for DOS, którego działanie opiera się na takiej właśnie koncepcji.

LAN WorkPlace for DOS v4.01

LAN WorkPlace for DOS (LWP) jest produktem programowym firmy Novell, instalowanym na stacjach roboczych IBM PC w sieci NetWare. Pakiet ten pozwala:

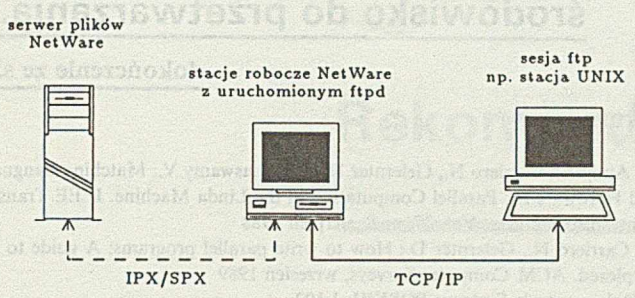
- pracować z systemem NetWare w taki sposób, jak przed zainstalowaniem pakietu,
- pracować z maszyną UNIX-ową na zasadzie terminala,
- uruchomić program umożliwiający swobodny transfer plików między maszyną UNIX-ową, serwerem NetWare i stacją roboczą IBM PC,
- pełnić rolę bramy (ang. *gateway*), przez którą użytkownicy systemu UNIX mogą uzyskać dostęp do plików serwera NetWare.

Przenoszenie plików

Usługi, jakich dostarcza pakiet LWP są wykonywane przez programy, których nazwy są podobne do tych, z jakimi mamy do czynienia w UNIX-ie. I tak do przesyłania zbiorów można użyć programu: ftp (*file transfer program*), *rcp* (*remote file copy*) lub *tftp* (*trivial file transfer program*). Programy te służą do kopiowania zbiorów lub całych kartotek. Oprócz kopiowania *ftp* pozwala na zmianę bieżącego dysku lub kartoteki oraz tworzenie i usuwanie plików i kartotek.

Wymienione programy działają w trybie *klient-serwer*, co oznacza, że udostępnienie usługi np. *ftp*, wymaga uruchomienia na maszynie UNIX-owej serwera tej usługi o nazwie *ftpd* (odpowiednio *tftpd* dla usługi *tftp*). Pakiet LWP jest również wyposażony w programy o tych samych nazwach *tftpd* i *ftpd*, które uruchomione na stacji roboczej IBM PC, czynią z niej serwera usług *ftp* lub *tftp*. Jakkolwiek w przypadku wielodostępnego UNIX-a klientami tych usług może być jednocześnie wielu użytkowników, to dostęp w danym czasie do IBM PC z jednozadaniowym DOS-em jest możliwy tylko dla jednego klienta.

Uruchamiając *ftpd* na stacji roboczej IBM PC, umożliwiamy użytkownikom UNIX-a także dostęp przez tę stację do plików serwera NetWare. IBM PC z *ftpd* pełni wówczas rolę dedykowanej bramy (ang. *gateway*) z takimi prawami dostępu, z jakimi uprzednio został nawiązany kontakt tej stacji z serwerem (rys. 1).



Rys. 1. Serwer *ftpd* uruchomiony na PC połączony równocześnie z serwerem plików NetWare, udostępnia dyski NetWare klientom *ftp*

Zdalne wykonywanie zleceń systemowych i emulacja terminala

rexec (*remote execution command*) i *rsh* (*remote shell*) to także programy typu *klient* służące do wykonywania zleceń systemowych na maszynie UNIX-owej. Ich odpowiednikami dla serwera są *rexecd* i *rshd*. Pakiet LWP nie zawiera takich serwerów, a więc nie można wykonać zdalnie komendy na innym IBM PC. Zlecenia, które można zdalnie wykonać, nie mogą wymagać interakcji ze strony użytkownika w czasie ich wykonywania, jak np. edycja zbioru.

Program *tnvt*220 wraz z rezydentnym *telapi* (*Telnet Application Programming Interface*) pozwala wykorzystać stację roboczą IBM PC do emulacji terminala. Możliwa jest emulacja terminali *DEC vt*220, *vt*100, *vt*52 lub terminala *ANSI*. Program ten pozwala na otwieranie kilku sesji terminalowych i uaktywnianie jednej z nich w określonym momencie.

Dodatkową usługą, którą udostępnia pakiet LWP jest możliwość drukowania na drukarkach przyłączonych do maszyn UNIX-owych. Polecenia pakietu LWP, pozwalające na wysłanie i usunięcie zadania z kolejki do drukarki, używają odrębnej składni. Dzięki plikom konfiguracyjnym są one tłumaczone na lokalne komendy maszyn UNIX-owych (pracujących pod systemami zgodnymi z UNIX BSD, System V, VMS z VMS/ULTRIX Connection).

Pakiet LWP i Microsoft Windows

Aplikacjami uruchamianymi pod systemem Microsoft Windows są: *Host Presenter, File Express* oraz *Serving FTP*, z których *Host Presenter* wraz z *telapi* jest odpowiednikiem emulatora terminala w tym środowisku, umożliwiając otwarcie do dziesięciu okien terminalowych. Podczas sesji emulatora istnieje możliwość kopiowania zawartości okien terminalowych do aplikacji Windows, takich jak *NOTEPAD* lub *WRITE*, jak również kopiowanie zaznaczonych obszarów tekstu między oknami terminalowymi. Nie istnieje jednak żadna integracja między sesją emulatora, a oknem dla systemu DOS. Środowisko w *Host Presenter* może być kształtowane przez samego użytkownika, a ewentualne zmiany są wtedy rejestrowane w *zbiorach parametrów*. Dodatkową opcją jest kontrola wykonywania sesji terminalowej przez opis działań, napisany w odrębnym języku, o nazwie *ScreenPlay*.

Program Serving FTP jest odpowiednikiem *ftpd*. Uruchomiony na stacji roboczej IBM PC z MS Windows czyni z niej niededykowany serwer usług *ftpd*. Z usług tych w danym czasie może korzystać tylko jeden klient. W danej chwili można również uruchomić jedynie jeden *Serving FTP*.

File Express zapewnia takie same usługi *ftp*, lecz w wydaniu dla klientów pracujących z MS Windows. Pracując z *File Express* uzyskujemy dostęp do plików jednej lub dwóch maszyn UNIX-owych i/lub stacji roboczych IBM PC z uruchomionym serwerem *ftp* (*ftpd* lub *Serving FTP*).

Wszystkie programy LWP dla DOS-a można uruchamiać w oknach MS-DOS pod Windows.

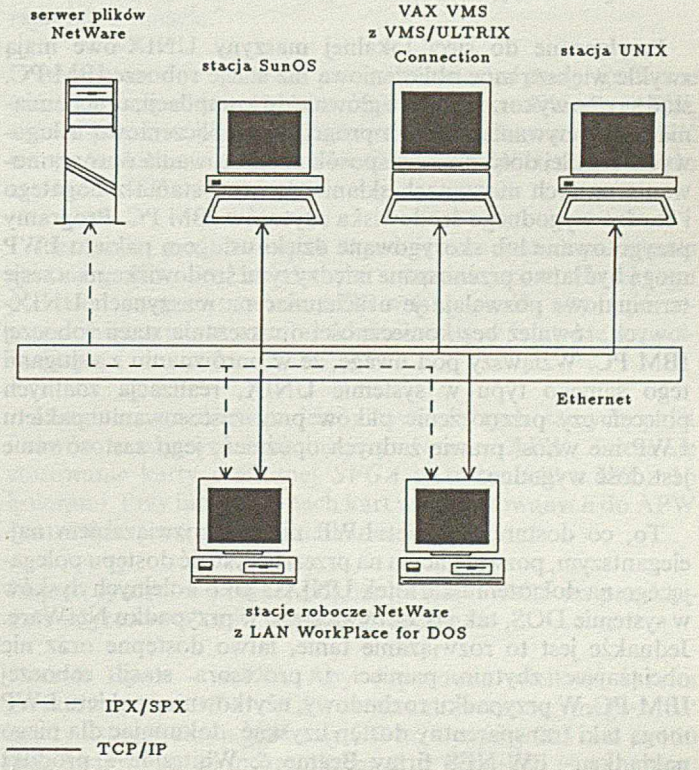
Wymagania pakietu LWP

Pakiet LWP, stosując dla DOS-a *ODI* (*Open Data Interface* – mechanizm pozwalający dzielić kartę sieciową i jej sterownik między takie protokoły, jak IPX/SPX i TCP/IP), udostępnia równolegle usługi świadczone przez serwera plików NetWare oraz usługi zawarte w pakiecie bez konieczności restartu komputera IBM PC.

Pakiet LWP zawiera drivery ODI dla kart 3Com, NE2, NE2000, NE1000 oraz Exos. Dodatkowe wymagania, jakie Novell stawia przy instalacji LWP, to: co najmniej komputer klasy IBM PC XT z systemem DOS od wersji 3.3. W przypadku użycia tylko programów uruchamianych pod systemem DOS, potrzeba 1,8 MB przestrzeni twardego dysku; dla pełnej instalacji, zawierającej programy dla DOS i Microsoft Windows v3.x jest konieczne 2,8 MB tej przestrzeni.

Uwagi eksploatacyjne

Do testowania pakietu LWP wykorzystano zarówno system NetWare, jak i maszyny UNIX-owe zainstalowane w Instytucie Informatyki Teoretycznej i Stosowanej PAN w Gliwicach. Fragment istniejącej przykładowej sieci, w której funkcjonuje pakiet LWP przedstawia rys. 2. Medium fizyczne – cienki Ethernet – łączy serwer NetWare v3.11 wraz ze stacjami roboczymi, stacje firmy Sun z systemem SunOS v4.1.x, kompu-



Rys. 2. Przykładowa sieć, w której LWP łączy DOS z systemami UNIX i VMS

ter VAX z systemem VMS v5.5 i VMS/ULTRIX Connection oraz komputery z systemem INTERACTIVE UNIX V/386 v3.2.

Oprogramowanie LWP można wprowadzać i usuwać do/lub z pamięci operacyjnej w dowolnym momencie pracy stacji roboczej, przy czym może być ono składowane w całości w serwerze NetWare i tym samym udostępniane w identyczny sposób wszystkim stacjom roboczym IBM PC. Do pracy współbieżnie na dwóch protokołach IPX/SPX oraz TCP/IP są potrzebne nakładki: LSL (*Link Services Layer*), ukierunkowująca pakiety od karty sieciowej do driverów odpowiednich protokołów, oraz MLID (*Multiple Link Interface Driver*), która odbiera i przesyła pakiety różnych protokołów do LSL. LSL i MLID zajmują łącznie 25 KB. Na to zostają załadowane drivery protokołów np. TCP/IP i IPX/SPX (razem 36 KB) – patrz rys. 3. Emulacja terminala pojedynczej sesji zajmie dodatkowe 190 KB, podobnie jak sesja *ftp*.

Rys. 3. Konfiguracja oprogramowania stacji roboczej pozwalająca na równoczesne korzystanie z protokołów TCP/IP oraz IPX/SPX

Przyłączone do sieci lokalnej maszyny UNIX-owe mają zwykle większą moc obliczeniową niż stacje robocze IBM PC, stąd są one wykorzystywane głównie do kompilacji, uruchamiania i wykonywania różnych programów obliczeniowo-usługowych. Z kolei dość żmudny sposób przygotowania oprogramowania na tych maszynach skłania do korzystania z bogatego i bardzo wygodnego środowiska edytorów IBM PC. Programy przygotowane lub skorygowane dzięki usługom pakietu LWP mogą być łatwo przenoszone między tymi środowiskami, a sesje terminalowe pozwalają je uruchamiać na maszynach UNIX-owych, również bez konieczności opuszczania stacji roboczej IBM PC. Wziąwszy pod uwagę, że w porównaniu z usługami tego samego typu w systemie UNIX, realizacja zdalnych poleceń czy przenoszenie plików przy zastosowaniu pakietu LWP nie wnosi prawie żadnych opóźnień, jego zastosowanie jest dość wygodne.

To, co dostarcza pakiet LWP nie jest rozwiązaniem najelegantszym, pozwalającym na przezroczystość dostępu polegającego na dołączeniu kartotek UNIX-a jako kolejnych dysków w systemie DOS, tak jak odbywa się to w przypadku NetWare. Jednakże jest to rozwiązanie tanie, łatwo dostępne oraz nie obciążające zbytnio pamięci i procesora stacji roboczej IBM-PC. W przypadku rozbudowy, użytkownicy pakietu LWP mogą taki transparentny dostęp uzyskać, dokupując dla niego nakładkę – BW-NFS firmy Beame & Whiteside – produkt polecany przez Novella – lub własny produkt Novella – NFS Client for LWP.

LITERATURA

[1] Novell Inc.: LAN WorkPlace for DOS Administrator's Guide.
[2] Udell J., Thompson T., Yager T.: Mix'n'match LAN. BYTE, listopad 1991.

LINDA
środowisko do przetwarzania...

dokończenie ze s. 5

LITERATURA

[1] Ahuja S., Carriero N., Gelernter D., Krishnaswamy V.: Matching Language and Hardware for Parallel Computation in the Linda Machine. IEEE Transactions on Computers, Vol. 37, nr 8, sierpień 1988
[2] Carriero N., Gelernter D.: How to write parallel programs: A guide to the perplexed. ACM Computer Surveys, wrzesień 1989
[3]Dokumentacja Systemu POSYBL-1.102
[4] Gelernter D.: Multiple Tuple Spaces in Linda. PARLE'89
[5] Gelernter D.: Parallelism to the People, or, Experience with Programmining in Linda
[6] Gelernter D., Philbin J.: Spending your Free Time. Byte, maj 1990
[7] Lerer W.: Linda Meets Unix. IEEE, luty 1990.

Uwaga!

INFORMATYKĘ już można zaprenumerować także na poczcie.

Szczegóły na II stronie okładki

meditronik SPÓŁKA z o.o.
• CZĘŚCI ELEKTRONICZNE
• KOMPUTERY PS/1,PS/2
• DRUKARKI HP
• INSTALACJE SIECI KOMPUTEROWYCH
Partnerzy handlowi: ANALOG DEVICES, ITT, MOTOROLA, SAMSUNG, TELEFUNKEN i inni

UMC UNITED MICROELECTRONICS CORPORATION
• UKŁADY PAMIĘCI
• UKŁADY KOMPUTEROWE
• UKŁADY KOMUNIKACYJNE I KOMERCYJNE
PRZEDSTAWICIELSTWO

HEWLETT PACKARD COMPONENTS
• TRANSOPTORY
• WSKAŹNIKI ŚWIETLNE
• WYŚWIETLACZE LED
• PRODUKTY KODÓW KRESKOWYCH
• KONTROLERY I CZUJNIKI RUCHU
• TECHNIKA ŚWIATŁOWODOWA
• ELEMENTY W.CZ. I MIKROFALOWE
• PODZESPOŁY DO MONTAŻU POWIERZCHNIOWEGO (SMD)
DYSTRYBUCJA

BOURNS
• POTENCJOMETRY TRIMPOT
• HYBRYDY REZYSTOROWE
• REZYSTORY SUBMINIATUROWE
• BEZPIECZNIKI MULTIFUSE
• POTENCJOMETRY PRECYZYJNE
• POTENCJOMETRY PANELI CZOŁOWYCH I KODERY
• CEWKI I TRANSFORMATORY
• CZUJNIKI CIŚNIENIA, POŁOŻENIA I PRZYŚPIESZENIA
PMI
DYSTRYBUCJA

IBM Business Partner

meditronik sp. z o.o.
00-194 Warszawa,ul. Dzika 4
tel. (02) 6352263, 6352264
fax (02) 6352195, tlx 816075
SO/575/91

DARIUSZ GAIK
ANDRZEJ PRZYBYLSKI
Zakład Elektroniki i Informatyki
Uniwersytet Śląski
Katowice

Rekomendacje narzędzi graficznych dla środowiska APW

Pakiet APW jest systemem autorskim środowiska MS Windows, przeznaczonym do łatwego i efektywnego projektowania w tym środowisku aplikacji demonstracyjnych (CAI – *Computer Assisted* Instruction). Do zalet APW zaliczyć można łatwość i efektywność tworzenia programów autorskich, a w szczególności:

- obiektowe ukierunkowanie aplikacji,
- bezpośrednia edycja aplikacji (edytor graficzny, edytor tekstu, zewnętrzny interfejs graficzny),
- możliwość projektowania animacji,
- zaawansowana analiza odpowiedzi,
- współbieżność procesów,
- tworzenie rozgałęzień programowych,
- tworzenie własnych zmiennych i modeli bibliotecznych,
- typowe instrukcje (jak w klasycznym języku),
- automatyczne tworzenie dokumentacji.

Programowanie w środowisku APW odbywa się przez deklarację sieci działań – algorytmu projektowanego programu. Sieć działań budowana jest wyłącznie z zestawu ikon środowiska APW.

Aplikacje tworzone za pomocą APW składają się zazwyczaj z dwóch części: prezentacyjnej i odpytującej.

W aplikacji APW realizacja części prezentacyjnej opiera się na:

- ikonach grafiki statycznej (ikona prezentacyjna i kasująca),
- ikonach grafiki dynamicznej (ikona animacyjna i oczekiwania),
- ikonie decyzyjnej i kalkulacyjnej.

Część odpytująca aplikacji APW wykorzystuje te same ikony oraz dodatkowo najważniejszą dla tej części ikonę interakcyjną. Tak zaprojektowany pakiet APW pozwala łatwo i szybko zaprojektować niewielką i niezbyt skomplikowaną aplikację dla MS Windows. Jednakże do projektowania skomplikowanych graficznie obiektów edytor APW wydaje się być zbyt ubogim i mało efektywnym narzędziem.

APW także nie zawiera żadnej biblioteki, choćby podstawowych typowych rysunków mogących stanowić podstawę do tworzenia własnych prezentacji graficznych, np. podstawowych symboli elektronicznych, elementów składowych rysunków mechanicznych itp. służących do prezentacji rysunków z różnych dziedzin wiedzy.

Badania realizowano w ramach projektu KBN nr 8-0275-91-01 i dofinansowano z programu badań własnych Uniwersytetu Śląskiego nr 8/IPT/92.

Biblioteki takie zawierają bardziej zaawansowane edytory graficzne, do których z pewnością można zaliczyć edytor graficzny *Corell Draw v.* 2.0, wraz z biblioteką dodatkowych fontów i biblioteką zawierającą tysiące rysunków podstawowych, o których wcześniej była mowa.

W tym celu użytkownik musi zatem sięgnąć do bardziej zaawansowanych edytorów graficznych. Na szczęście wbudowany w APW importer grafiki umożliwia pobieranie grafik wektorowych i typu mapy bitowej (ang. *bit-mapp*) w postaci plików *.PCX, .PNT, .PIC, .PCT, .WMF, .BMP, .DIB.*

Corell Draw nie jest oczywiście jedynym dostępnym edytorem graficznym i dlatego zostanie porównany z innymi edytorami graficznymi, takimi jak *Paint Brush* oraz *Designer*. Jednak główny ciężar położono na edytor *Corell Draw* mający największe możliwości graficzne oraz możliwość eksportu grafiki w formacie. *WMF* (*Windows Meta File*) i *.PCT*, które to formaty gwarantują import grafiki do APW w postaci zestawu obiektów graficznych. Taki format grafiki w niektórych zastosowaniach jest nieodzowny do tworzenia prezentacji zawierających animacje.

Wykorzystując bardziej rozbudowane edytory graficzne (edytory uzupełniające) do projektowania grafiki dla APW, należy przed rozpoczęciem czynności projektowania zwrócić uwagę na kilka zagadnień związanych z importem grafiki do systemu APW.

Pierwszym podstawowym zagadnieniem jest dopasowanie ekranu graficznego w edytorze uzupełniającym, do ekranu prezentacyjnego w systemie APW w taki sposób, aby importowane obiekty miały rozmiar zdefiniowany w edytorze uzupełniającym.

Drugim zagadnieniem jest ustawienie tych samych palet kolorów w edytorze uzupełniającym oraz w APW. Zaleca się stosowanie karty graficznej *SVGA* w trybie 640 × 480 z 16 kolorami. Przy innych trybach kart w importowanych do APW rysunkach następują przekłamania kolorów.

Import grafiki w postaci plików .WMF i .PCT na przykładzie edytorów Corell Draw i Designer

Correll Draw – APW. Aby dopasować wielkość ekranu graficznego *Corell Draw* (CD) do ekranu prezentacyjnego APW należy ustawić ekran graficzny CD na wymiary 225 mm × 160 mm. Ustawienie to można dokonać w opcji CD Custon Size. W pakiecie APW należy jednocześnie pamiętać, że importowana grafika będzie umieszczona w punkcie centralnym

równym bieżącej pozycji kursora graficznego. Należy podkreślić, że tego typu skalowanie jest możliwe tylko dla eksportu grafiki typu wektorowego. Dla grafiki typu *bit-map* nie można podać jednoznacznej zależności, gdyż rozmiary rysunku zależą od przyjętej rozdzielczości rysunku. Skalowanie to ma zależność liniową.

Następne zagadnienie, to ustawienie tych samych palet kolorów w *CD* i APW (sugeruje się, że pożądane efekty uzyskuje się z kartami graficznymi SVGA pracującymi w trybie 640 × × 480 z 16 kolorami). Przy innych trybach kart następują bliżej nie określone przekłamania kolorystyki importowanych rysunków do okien prezentacyjnych APW.

Nie zaleca się także stosowania nadmiaru efektów specjalnych, a w szczególności efektów typu *Extrude* (dodanie efektów pozornej trójwymiarowości), który to efekt dodaje do podstawowego rysunku bardzo dużą liczbę obiektów widocznych w APW, z którymi pakiet ten „nie może sobie poradzić", kończąc pracę komunikatem informacyjnym o za dużej liczbie obiektów do importu grafiki.

Zaleca się przygotowanie rysunków prawie całkowicie gotowych do importu przez APW, ponieważ ewentualne poprawki grafiki zaimportowanej do APW kończą się zwykle zupełnym niepowodzeniem z powodu innego zapisu obiektu w edytorze uzupełniającym niż w APW. Np. zwykły okrąg w CD jest jednym bardzo łatwo modyfikowanym obiektem. W środowisku APW staje się on czterema obiektami (cztery łuki 90°). Wypełniony prostokąt jest jednym obiektem w CD, natomiast w APW staje się już pięcioma obiektami (cztery linie brzegowe oraz wypełnienie). Tak zdefiniowane obiekty za pomocą edytora APW można w łatwy sposób jedynie przemieszczać i zmieniać ich kolory. Próba zmiany ich wielkości jest dość kłopotliwa i zazwyczaj kończy się niepowodzeniem.

Narzędzia graficzne APW w zasadzie powinny jedynie służyć do pewnej „kosmetyki" graficznej przygotowywanego okna. Należy przypomnieć w tym miejscu, że APW ma tę cenną cechę, że kolejne grafiki z ikon prezentacyjnych mogą się na ekranie nakładać kolejno. Z tego powodu wydaje się korzystniejszym przygotowywanie rozbudowanych prezentacji w kilku etapach. Kolejną niedogodnością jest fakt niemożności eksportu poprawionej grafiki powtórnie do CD. Przedstawione ograniczenia dotyczące importu grafiki wektorowej z edytora CD nie dyskwalifikują współpracy obu tych programów przy tworzeniu rozbudowanej grafiki prezentacyjnej, uwzględniając przede wszystkim wspomnianą już wcześniej bibliotekę symboli i fontów graficznych.

Designer – APW. Edytor Designer ma także możliwość eksportu grafiki *.WMF* i *.PCT*. Wyżej podane uwagi na temat *Corell Draw* w zasadzie dotyczą także i tego edytora. Niemniej istnieje istotna różnica dotycząca wielkości importowanych rysunków dla obu tych edytorów. I tak „edytując" na pełnym ekranie z wykorzystaniem opcji *Actual size*, po czynnościach eksport-import do APW otrzymujemy rysunek wypełniający okno prezentacyjne APW zaledwie w kilku procentach. Pełne zajęcie okna prezentacyjnego w APW, to w *Designer* rysunek o wymiarze 300 mm × 240 mm. Dodatkowo, przy imporcie grafiki *.PCT* należy ustawić *Scale Factor* na 55%. Ustawienie innej wartości spowoduje odpowiednio zwiększenie lub zmniejszenie gabarytów obiektu na ekranie prezentacyjnym APW. Aby uzyskać tę samą paletę kolorów, należy dodatkowo ustawić druga wersję *PICT* w oknie *PICT Output*.

Import grafiki bit-mapowej

Wydaje się, że tego typu grafikę najkorzystniej można tworzyć używając prostego edytora graficznego *Paint Brush* (PB). Podobne zagadnienie to wielkość rysunków w PB oraz w APW. Podstawowy ekran w PB (540 × 360 piksli) odpowiada ok. 90% powierzchni ekranu okna prezentacyjnego w APW. Rysunki zaimportowane w postaci plików *.BMP* można oczywiście przeskalowywać, ale i tu wydaje się, że najkorzystniej jest je przygotować, możliwie w sposób ostateczny, przez program PB. Należy podkreślić, że eksport grafiki należy przeprowadzać za pomocą opcji *Copy to* po zaznaczeniu żądanego fragmentu w PB oraz ustawić w opcji *Options* odpowiednią liczbę kolorów, co oczywiście wiąże się bezpośrednio z wielkością tworzonego pliku.

W tworzonym oknie prezentacyjnym APW istnieje możliwość „mieszania" importu grafiki wektorowej i typu bit-map.

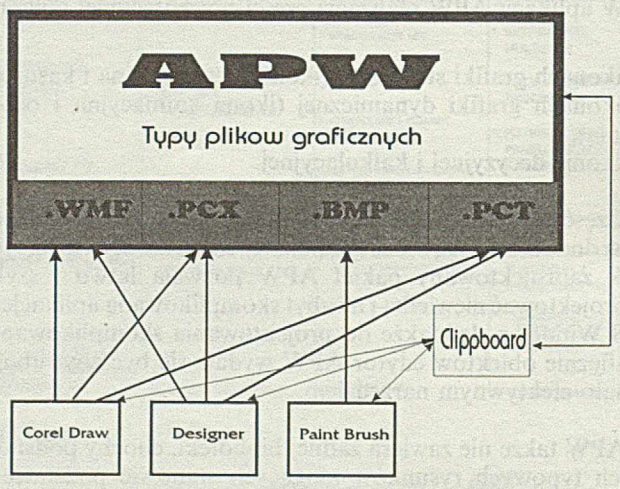
Import grafiki przez pliki .PCK

Jest to kolejny sposób importu grafiki *bit-map* z edytora *Corell Draw* CD. Należy stosować ten sposób importu przy mocno skomplikowanych rysunkach z dużą liczbą obiektów (szczególnie po zastosowaniu efektów specjalnych w CD). Import grafiki jest monochromatyczny z dużą liczbą szczegółów przy rozdzielczości *coarse*-40 oraz *low*-70. Wraz ze zmianą rozdzielczości zmieniają się oczywiście wielkości rysunku w APW. Ten sposób importu grafiki zaleca się wyłącznie do grafik jednobarwnych (kreskowych).

Import grafiki przez Clippboard

Przenoszenie rysunków tym sposobem jest możliwe zarówno w postaci grafiki typu *bit-map*, jak i wektorowej. Jest to najprostszy i najszybszy rodzaj importu grafiki do APW. Ponadto wielkość rysunku jest dokładnie taka sama, jak w zastosowanym edytorze. Wadą tej metody jest brak wzorca obiektu w postaci pliku, co bardzo skomplikuje odtworzenie obiektu w razie popełnienia błędu w czasie projektowania.

Clippboard jest jedynym sposobem eksportu grafiki z APW do innych edytorów, ale i tu należy liczyć się ze zniekształceniami rysunków. W przypadku dużej liczby obiektów (stopnia skomplikowania rysunku) podobnie i tu liczba ok. 50 obiektów stanowi rozsądną granicę importu.



Możliwości eksportu grafiki do APW

Na rysunku przedstawiono wyżej opisane sposoby eksportu grafiki do APW w postaci plików *.WMF, .PCX, .BMP, .PCT* oraz przez *Clippboard*.

dokończenie na s. 21

JAN GOLIŃSKI
Szkoła Główna Handlowa
Warszawa

Problemy projektowania systemów informacyjnych

Nie wiem co to takiego analiza systemowa i nie rozpoznałbym jej, choćby mi ją położono na kolanach, ale wiem, że ilekroć przedkłada mi się budżet opracowany metodą systemową, to koszt jest trzykrotnie wyższy od spodziewanego.
Admirał Hyman G. Rickover

W krajach o wysokim poziomie informatyki można zauważyć od wielu już lat istotne rozgraniczenie zadań między osobami personelu tworzącego systemy przetwarzania danych. Początkowo często łączono zadania projektanta systemu i programisty, ponieważ w wielu przypadkach była to ta sama wszechstronnie wykształcona osoba. Przez wykształcenie takie rozumieć należy ukończone studia wyższe, jak również kursy projektowania i programowania. Z biegiem czasu nastąpiły tu istotne zmiany. Programowanie, ze względu na dostępne narzędzia, prostotę i elastyczność nowych języków programowania staje się coraz łatwiejsze. Stąd coraz wyraźniejsze rozgraniczenie zadań przy tworzeniu systemów informacyjnych. Programowaniem zajmują się osoby o coraz mniejszych kwalifikacjach w sensie ogólnym – staje się ono zajęciem dla ludzi coraz młodszych. Inaczej przedstawia się sytuacja z analitykami systemu. Ich rola systematycznie rośnie. Systemy informatyczne zaczęły bowiem obejmować coraz bardziej skomplikowane dziedziny funkcjonowania wielkich przedsiębiorstw.

W większości krajów europejskich, prawie każdy młody człowiek z wyższym wykształceniem uzyskuje w czasie studiów minimum wiedzy w zakresie ogólnosystemowym i informatycznym. Jest to na ogół wystarczająca podstawa, pozwalająca analitykowi systemu na rozpoczęcie działalności, oczywiście z założeniem, że musi on być permanentnie kształcony i informowany o nowościach zachodzących w zakresie sprzętu informatycznego i oprogramowania. Wynika stąd również konieczność studiowania właściwego piśmiennictwa, którego tak wiele na półkach księgarskich. W polskim piśmiennictwie technicznym znaleźć można sporo pozycji z zakresu programowania, często na bardzo wysokim, wręcz znakomitym poziomie. Są to monografie zarówno polskich autorów, jak i świetnie wybrane przekłady literatury światowej. Gorzej wygląda sprawa z wydawnictwem z zakresu projektowania systemów. Krajowych pozycji brak, natomiast kryteria doboru tłumaczeń z literatury zagranicznej są mało klarowne.

W miarę tworzenia coraz bardziej kompleksowych systemów przetwarzania informacji, na co istotny wpływ miał m.in. rozwój astronautyki, udział kosztów oprogramowania w łącznych kosztach ponoszonych na informatykę zaczął gwałtownie wzrastać. Już w 1980 r. na oprogramowanie wydano w RFN 8 mld marek, a w USA – ponad 10 mld dolarów, co stanowiło ok. 1% dochodu narodowego brutto. Dzisiaj jest to już dwukrotnie więcej. Jednocześnie coraz bardziej wzrastało niezadowolenie z niewystarczającej jakości oprogramowania. Tłumaczono to często tym, że programowanie jest młodą dyscypliną, nie dysponującą metodami znanymi w innych, liczących dziesiątki a nawet setki lat dziedzinach wiedzy.

Ostatnio coraz wyraźniej jest jednak widoczne dążenie do racjonalizacji procesu wytwarzania oraz polepszenia jakości oprogramowania. Coraz częściej kierowano uwagę na dyscypliny inżynierskie oraz stosowane tam od długiego czasu dobrze sprawdzone metody i techniki. Był to początek nowej dyscypliny, dla której przyjęła się nazwa inżynierii oprogramowania (ang. *software engineering*).

Dotychczas przyjmowano średni okres użytkowania produktu programowego na ok. 7 lat, przy czym stosunek kosztów realizacji produktu do kosztów jego utrzymania i eksploatacji kształtował się jak 1 do 2. A więc produkt, który kosztował 100 mln zł, „skonsumuje" dodatkowo w ciągu swego siedmioletniego cyklu życia jeszcze 200 mln zł.

Systemy stały się bardziej kompleksowe, co oznacza znaczny wzrost kosztów inwestycyjnych. Dlatego też obecnie realizowane systemy zakładają okres życia produktu programowego na 12–15 lat. Łatwo zatem oszacować, że w związku z takim założeniem jeszcze zwiększy się udział kosztów jego konserwacji użytkowania. Jednocześnie z upływem lat oprogramowanie staje się bardziej kosztowne w użytkowaniu. Już dzisiaj można spotkać coraz większą liczbę przedsiębiorstw, które muszą przeznaczać 80% potencjału swego personelu informatycznego na konserwację eksploatowanych systemów. Tendencja ta ma charakter rosnący. Wiele przedsiębiorstw nie będzie miało prawie żadnej możliwości rozwijania nowych systemów bez istotnego zwiększenia zatrudnienia projektantów i programistów. Stąd też znacznie wzrasta zapotrzebowanie na dobrze wykształconych ludzi, zwłaszcza o profilu analityka. Ocenia się, że np. USA z końcem bieżącego stulecia będą miały w tej dziedzinie deficyt, wyrażający się liczbą ok. 2 mln projektantów i programistów.

Podstawowym celem w dziedzinie inżynierii oprogramowania jest stworzenie produktu, który można łatwo eksploatować i konserwować, a dopiero w drugiej kolejności rozpatruje się problem optymalizacji nakładów na fazę tworzenia systemu. Trzecim w kolejności celem jest zwiększenie niezawodności

systemu, głównie przez eliminację błędów. Wreszcie jako czwarty cel należy wymienić wymóg przenośności (portabilności) oprogramowania użytkowego (aplikacyjnego). Ten ostatni wymóg jest szczególnie istotny dla wielkich przedsiębiorstw, które w wielu krajach mają liczne ośrodki obliczeniowe i najczęściej różny sprzęt.

Wyżej wymienione i oczywiście inne jeszcze problemy składają się na zjawisko, określane często „kryzysem softwarowym", który wydaje się być daleki od zakończenia. Kryzys ten można sprowadzić do kilku lapidarnych pytań:

- Dlaczego realizacja systemów i programów trwa za długo?
- Dlaczego oprogramowanie jest tak kosztowne?
- Dlaczego z taką trudnością i tak długo wykrywa się błędy w programach, a mimo to programy nadal przekazuje się użytkownikowi z błędami?
- Dlaczego istnieją trudności w określaniu postępów prac związanych z realizacją systemów i z czego one wynikają?

Cała ta wielokrotnie dyskutowana problematyka jest wciąż aktualna. Jej rezultatem jest kilka mitów, krążących w środowisku informatycznym, niezależnie od szerokości geograficznej. Spróbujmy je pogrupować wg źródeł powstawania oraz porównać z konkretną rzeczywistością.

Mity kierownictwa

Mit 1: Dlaczego mamy zmienić nasz stosunek do problemu tworzenia oprogramowania? Czy nie powinniśmy dzisiaj tworzyć ten sam rodzaj oprogramowania i w taki sam sposób, jak przed dziesięciu laty?

Rzeczywistość: Obszar zastosowań być może pozostał ten sam, ale dla wielu organizacji zmieniły się nader istotnie wymagania. Dzisiaj dla osiągnięcia strategicznych celów przedsiębiorstwa znacznie ważniejszymi celami są większa efektywność oraz jakość oprogramowania.

Mit 2: Mamy przecież wystarczające standardy i procedury postępowania związane z oprogramowaniem.

Rzeczywistość: Być może jest tak istotnie, ale czy z tych norm korzystamy? Czy praktycy są świadomi ich istnienia? Czy standardy te odzwierciedlają współczesne tendencje w rozwoju oprogramowania? W wielu przypadkach odpowiedź na to pytanie jest negatywna.

Mit 3: Często kierownictwo posługuje się następującym argumentem: Ludzie dostali narzędzia odpowiadające aktualnemu poziomowi technologii – zainstalowano im przecież nowy komputer. O co więc w tym wszystkim chodzi?

Rzeczywistość: Produkowanie oprogramowania wysokiej jakości wymaga znacznie więcej niż zainstalowanie nowego komputera. Dla uzyskania odpowiedniej jakości produktu programowego ważniejsze niż nowa instalacja sprzętu komputerowego są, na przykład, narzędzia ułatwiające produkcję oprogramowania.

Mit 4: Jeżeli zagraża nam niedotrzymanie terminów, to aby dotrzymać zobowiązań można po prostu zwiększyć liczbę programistów (niektórzy nazywają to „koncepcją mongolskiej hordy").

Rzeczywistość: Produkcja oprogramowania wciąż jeszcze nie jest, tak jak większość rodzajów produkcji przemysłowej, procesem automatycznym. Brooks pisze: *...adding people to a late software project makes it later* (dodawanie ludzi do opóźnionego projektu opóźnia go jeszcze więcej). Na pierwszy rzut oka wydaje się, że teza ta przeczy naszej intuicji. Jednakże wprowadzenie nowych ludzi redukuje czas na rozwój systemu, ponieważ powstaje konieczność ich przyuczenia oraz włączenia w istniejący system komunikacji. Do projektu można dodawać ludzi tylko w sposób przemyślany i skoordynowany.

Mity użytkownika

Mit 1: Uważa się często, że do pisania programów wystarcza tylko ogólne określenie celów, a szczegóły można uzupełnić później.

Rzeczywistość: Złe lub ubogie zdefiniowanie zadania w początkowej fazie projektu jest główną przyczyną wadliwego produktu końcowego. Formalny i dokładny opis obszaru informacji, funkcji, wydajności, sprzężeń, ograniczeń konstrukcyjnych oraz kryteriów istotności są niezmiernie ważne. Można to uzyskać tylko przez bezpośrednią i ciągłą komunikację oraz współpracę z użytkownikiem.

Mit 2: Wymagania projektowe wprawdzie zmieniają się, ale łatwo to uwzględnić, ponieważ oprogramowanie jest z natury rzeczy elastyczne.

Rzeczywistość: Prawdą jest, że wymagania dotyczące produktu programowego zmieniają się, lecz konsekwencje tych zmian są zależne od czasu, kiedy zmiany te wprowadza się. Jeżeli następuje to w fazie definicyjnej, to konsekwencje są stosunkowo nieznaczne (podwojenie pracochłonności). W fazie drugiej realizacji projektu wpływa to już w istotnym stopniu na koszt realizacji (1,5–6 razy większy), a kiedy zmiany te przeprowadza się w fazie eksploatacji, mogą one spowodować szokujące przekroczenie przewidywanych na początku kosztów (60–100-krotne).

Mity informatyka – praktyka

Mit 1: W rzeczywistości brak jest takich metod analizy, projektowania oraz testowania, które działałyby prawidłowo i efektywnie. Dlatego nadal wydaje się, że najlepsze jest postępowanie według zasady: „Siadam przy terminalu i zaczynam programować".

Rzeczywistość: Sprawdzone metody analizy, projektowania, testowania są dzisiaj już bardzo szeroko stosowane. Żadne z nich nie są jeszcze idealne w działaniu praktycznym. Natomiast metodologie są dostępne dla wszystkich.

Mit 2: Uważa się, że jeżeli program został napisany i działa poprawnie, to programista zrobił wszystko, co do niego należało.

Rzeczywistość: Znane jest porzekadło: „Im wcześniej zaczynasz pisać program, tym później będziesz miał gotowy produkt programowy".

Mit 3: Powszechny wśród programistów, a sprowadzający się do przekonania, że dopóki program nie „chodzi", nie ma czasu na zajmowanie się problemem jego jakości.

Rzeczywistość: Jednym z podstawowych filtrów, umożliwiających uzyskanie dobrego oprogramowania jest formalna i techniczna kontrola każdej przeprowadzonej fazy realizacji zadania. Stwierdzono, że daje to więcej, niż testowanie napisanych już programów.

Mit 4: Jedynym sprawdzianem dobrego projektu jest funkcjonujący program.

Rzeczywistość: Funkcjonujący program jest tylko jednym z elementów „konfiguracji oprogramowania". Dokumentacja stanowi podstawę dobrego produktu oraz – co jest ważniejsze – pozwala sterować zadaniami eksploatacyjnymi.

Mit 5: Jeżeli oprogramowanie jest w stałym użyciu, konserwacja jest minimalna i może być przeprowadzana w dowolny sposób.

Rzeczywistość: Na konserwację planowo przeznacza się stosunkowo niewielki procent budżetu, ale w rzeczywistości (co stanowi dla wielu szefów duże zaskoczenie) na utrzymanie oprogramowania z reguły ponosi się więcej kosztów niż pierwotnie zaplanowano. Dlatego też konserwacja (pielęgnacja) oprogramowania powinna być starannie planowana, organizowana i kontrolowana. Problem ten należy traktować jako szczególnie ważny element.

* * *

Celem przemysłowej produkcji oprogramowania jest osiągnięcie wysokiej efektywności i jakości przy utrzymaniu planowanych terminów i kosztów. Cele te tylko wtedy dadzą się zrealizować, gdy główne elementy procesu tworzenia oprogramowania: produkcja, kontrola jakości, kierowanie oraz jego konserwacja będą sensownie wsparte odpowiednią metodyką oraz właściwymi narzędziami programowymi.

Rekomendacje narzędzi graficznych dla środowiska APW

dokończenie ze s. 18

Przedstawione w artykule uwagi są wynikiem eksperymentów autorów, jakie zebrano w trakcie oceny cech użytkowych pakietu APW. Oprogramowanie pozyskano za pośrednictwem Tempus JEP 1087. Badania realizowano w Zakładzie Elektroniki i Informatyki we współpracy z Hamburger Ausbildungspartnershaft w Hamburgu. Prace są kontynuowane, ze szczególnym uwzględnieniem systemów multimedialnych, współpracujących z APW.

LITERATURA

[1] Draus A., Mazur T.: CorellDraw wersja 2.0, PLJ, Warszawa 1991
[2] Gaik D., Kotula J., Piecha J.: System autorski ASYS. INFORMATYKA 5/1992, str. 17–23
[3] Piecha J.: Systemy CAT. INFORMATYKA nr 4, 1991, s. 1–6
[4] Piecha J., Przybylski A., Zyguła J.: APW – graficzne narzędzia programowania. INFORMATYKA nr 6, 1992, s. 20–26.

ECHO komputerowy serwis informacyjny

dokończenie ze s. 13

LITERATURA

[1] CCL Training Manual. Commission of The European Communities. Luxembourg, OPOCE 1991
[2] Echo Databases And Services. Wyd. 6, Commission of The European Communities, Luxembourg, OPOCE 1991
[3] I'M Guide – Information Market Guide, Commission of The European Communities. Luxembourg, IMPACT 1991
[4] User Manual. Commission Of The European Communities. Luxembourg, ECHO 1989.

Kongres IFIP'94

Dotarła już pierwsza oficjalna informacja na temat kolejnego, 13. Światowego Kongresu IFIP. Ta od przeszło 30 lat najważniejsza w skali światowej naukowa konferencja informatyczna odbywa się obecnie w cyklu dwuletnim. Jak wskazuje nazwa, organizatorem Kongresu jest Międzynarodowa Federacja Przetwarzania Informacji IFIP (International Federation for Information Processing), powołana przez UNESCO i skupiająca obecnie profesjonalne organizacje informatyczne ok. 70 krajów.

IFIP'94 odbędzie się w Hamburgu, po 32 latach ponownie na terenie RFN (w 1962 r. Monachium). Współorganizatorem kongresu jest reprezentant gospodarzy – niemieckie towarzystwo informatyczne GI (Gesellschaft für Informatik), założone w 1969 r. i skupiające obecnie ponad 20 000 członków. Obrady Kongresu będą odbywały się w hamburskim Centrum Kongresowym, jednym z najbardziej nowoczesnych i największych obiektów tego typu w Europie, w terminie **od 28 sierpnia do 2 września 1994 r.**

Hasłem przewodnim Kongresu będzie „Ewolucja komputera i telekomunikacji oraz jej siły napędowe" (Computer and Communications Evolution – The Driving Forces). W haśle tym zawarte są następujące pytania, na które powinny odpowiedzieć obrady kongresu:

- czy rozwijamy właściwą technologię i zastosowania?
- jakie będzie oddziaływanie tego rozwoju?
- co powinniśmy zmienić?

Przewodniczącym kongresu będzie nestor niemieckich informatyków prof. Wilfried Brauer z Politechniki Monachijskiej, przewodniczącym Komitetu Organizacyjnego – prof. Karl Kaiser z Uniwersytetu Hamburskiego, a przewodniczącym 35-osobowego Komitetu Programowego – dr Ronald Uhlig z firmy Northern Telecom (USA). Wśród członków komitetu programowego zauważyć można nazwiska dwóch polskich informatyków: Leszka Pacholskiego z Uniwersytetu Wrocławskiego oraz Andrzeja Salwickiego z francuskiego University de Pau, poprzednio Uniwersytetu Warszawskiego.

Termin zgłaszania referatów upływa **14 stycznia 1994 r.** Zgłoszenia będą sukcesywnie oceniane, a decyzje o przyjęciu przesyłane do połowy marca 1994. Termin dostarczenia zaakceptowanej wersji referatu wyznaczono na **22 kwietnia 1994 r.** Mimo, że do upływu terminu zgłaszania referatów pozostało jeszcze pełne 8 miesięcy, pragniemy zachęcić potencjalnych autorów w kraju, zarówno pracowników nauki, jak i przedstawicieli praktyki, do licznego zgłaszania referatów na tę szczególnie prestiżową konferencję, aby stworzyć liczniejszą, niż na poprzednich kongresach IFIP, reprezentację naszego kraju.

W PRZEDPŁACIE DUŻO TANIEJ

W II kwartale br. ukaże się na rynku drugie już wydanie **„Słownika skrótów angielskich stosowanych w elektronice, informatyce i telekomunikacji"**, najobszerniejsze, zawierające ok. 10 000 haseł, nowoczesne i bardzo poszukiwane, o czym świadczy fakt, że pierwsze wydanie rozeszło się błyskawicznie! Słownik można otrzymać pocztą pod wskazany adres, wpłacając preferencyjną, taką samą jak przy pierwszym wydaniu (pomimo inflacji!), kwotę 50 000 zł na konto:

PBK S.A. III O/Warszawa nr 370015-1573-139-11
Wydawnictwo SIGMA-NOT Sp. z o.o.,
Zakład Kolportażu
00-950 Warszawa, skr. poczt. 1004

Zamawiając drugą edycję słownika w przedpłacie, ma się gwarancję otrzymania go po cenie ulgowej, bez wysokiej marży sieci handlowej i kosztów wysyłki.

The Third International Conference Information Systems Developers Workbench Sopot 22–24 września 1992 r.

W dniach 22–24 września 1992 r. odbyła się w Sopocie „The Third International Conference on Information Systems Developers Workbench" (Trzecia Międzynarodowa Konferencja na temat warsztatu projektanta systemów informatycznych). Organizatorem tej imprezy była Katedra organizacji przetwarzania Danych Uniwersytetu Gdańskiego. Przedmiotem tej trzeciej z kolei cyklicznej Konferencji były metody, techniki i narzędzia tworzenia systemów informatycznych w organizacjach gospodarczych i administracyjnych, prezentacja praktycznych rozwiązań oraz zastosowań osiągnięć teoretycznych w tej dziedzinie. Konferencja ubiegłoroczna była przedłużeniem idei „The First Polish-Scandinavian Seminar on Current Trands in Information Systems Development Metodologies" (Paraszyno 1988), jak również „The Second International Conference on Information Systems Developers Workbench", która odbyła się w Sopocie w 1990 r. W trakcie obydwu tych konferencji wielu uznanych w świecie naukowców zaprezentowało swoje referaty.

W ciągu kilku ostatnich lat nastąpił rozwój już istniejących oraz powstanie nowych koncepcji tworzenia systemów informatycznych. Do tych zjawisk można zaliczyć inżynierię informacyjną, narzędzia CASE, języki czwartej generacji, systemy eksportowe, systemy obiektowo zorientowane i inne. Również w Polsce nowoczesne metody tworzenia systemów informatycznych niezbędnych do prawidłowego funkcjonowania organizacji gospodarczych i administracyjnych, budzą coraz większe zainteresowanie, zwłaszcza ostatnia z odbywających się co dwa lata Konferencji cieszyła się zainteresowaniem polskich informatyków, także spoza kręgu pracowników naukowych wyższych uczelni. O rosnącej randze cyklu konferencji świadczy udział uznanych autorytetów, takich jak T. W. Olle, A. Nilsson, F. Stowell, B. Sundgren, I. Hawryszkiewycz. Międzynarodowy Komitet ubiegłorocznej Konferencji liczył 51 osób z wielu krajów Europy Zachodniej, Centralnej, USA oraz Australii. W Konferencji brało udział ok. 100 uczestników z 15 krajów, zostało wygłoszonych 41 referatów. Konferencja była podzielona na sesje obejmujące 3–5 referatów. Ze względu na ograniczony czas, część z nich odbywało się równolegle. Poniżej przedstawiono spis sesji wraz z charakterystykami typowych referatów. Wszystkie referaty w języku angielskim zawarte są w materiałach wydanych w formie książki i są dostępne w Katedrze OPD Uniwersytetu Gdańskiego, Sopot, ul. Armii Krajowej 119/121, tel. (0-58) 51-00-61 wew. 400.

Sesje plenarne

Referat otwierający konferencję pt. „Information Systems Development – Implication for Education and Training" wygłosił F. A. Stowell (University of Paisley, Szkocja). Autor omówił zmiany jakie nastąpiły w dziedzinie tworzenia systemów informatycznych w ciągu ostatnich lat, m.in. rozwinięcie się specjalności *Analiza Systemów Informatycznych*. Profesjonaliści z tej dziedziny spotykają się w swej pracy z użytkownikami coraz lepiej znającymi zagadnienia informatyki i stawiającymi coraz większe wymagania. Powoduje to potrzebę zmian posiadanych już umiejętności w kierunku uwzględnienia nie tylko zmian w technologii, lecz również w organizacji i zarządzaniu. Analitycy systemów informatycznych muszą obecnie w coraz większym stopniu patrzeć na systemy całościowo, a nie tylko jako na systemy przetwarzania danych. Wszystko to oznacza, że powinni być oni wyposażeni w nowe umiejętności do rozwiązywania zarówno problemów technicznych, jak i organizacyjnych. Autor w swoim referacie omawia koncepcyjne oraz organizacyjne zagadnienia edukacji i szkolenia specjalistów z dziedziny systemów informatycznych.

Ostatnio w dziedzinie tworzenia systemów informatycznych zachodzą duże zmiany. Metody dotąd używane przestają zakładać jako jeden z pewników, że cele działania systemu są dobrze zdefiniowane od samego początku. Tego typu sytuacje dokładnie omówił B. Sundgren (Stockholm School of Economics, Szwecja) w referacie „Designing Information Systems with Partially Unknown Purposes". Została wskazana i zdefiniowana kategoria systemów o niesprecyzowanych na początku celach, takich jak np:

- systemy wspomagania decyzji (ang. *DSS*),
- wykonawcze systemy informatyczne (ang. *Executive information systems*),
- naukowe systemy informatyczne,
- systemy metainformatyczne.

Dopiero w trakcie tworzenia systemu tego typu pojawiają się całkowicie nowe jego zadania, które trzeba na bieżąco uwzględniać. W referacie przedstawiono wymagania i wyzwania stawiane nowym metodom rozwiązywania tych problemów. W dodatku do referatu zamieszczono streszczenie ćwiczeń analizy przypadku systemu metainformatycznego o częściowo określonych celach.

W trakcie trzeciej sesji plenarnej przedstawiono dwa referaty:

- dr A. Florczyk (Dyrektor Generalny Biura Do Spraw Informatyki Urzędu Ministrów) – „Strategic Application of Information Systems nad Telecommunication in Governmental Administration",
- dr J. B. Czermiński (Pełnomocnik Rektora Do Spraw Informatyki na Uniwersytecie Gdańskim) – „Metropolitan Academic Computer Network in University of Gdańsk".

Sesja „Strategiczne planowanie systemów informatycznych"

Prof. A. G. Nilsson (Stockholm School of Economics, Szwecja) w referacie pt. „Business Modelling as a Base for Information Systems Development", zaproponował wykorzystanie istniejących ogólnych metodyk modelowania procesów gospodarczych do procesów tworzenia systemów informatycznych w przedsiębiorstwach. Analitycy i projektanci systemów zaangażowani w stosowanie istniejących metodyk tworzenia systemów informatycznych nie są świadomi istnienia wielu podejść do modelowania proce-

sów gospodarczych. Referat podejmuje ten temat w aspekcie teoretycznym i empirycznym na podstawie przeprowadzonych przez autora eksperymentów z koordynacją tych dwóch rodzajów metod.

Sesja „Teoretyczne zagadnienia systemów informatycznych"

Wyniki obszernego badania ankietowego kluczowych zagadnień zastosowania informatyki w gospodarce i administracji przedstawili S. Dekleva (DePaul University, Chicago, USA) oraz J. Zupančič (University of Maribor, Słowenia) w referacie pt. „An Empirical Investigation of Key Issues in Information System Management". W badaniach ankietowych uczestniczyło 200 kierowników działów informatycznych w Słowenii. Określili oni, które z 25 najbardziej istotnych zagadnień będą istotne w najbliższych 3–5 latach. Badania realizowano metodą delficką w kolejnych trzech rundach. Wyniki porównano z analogicznymi opracowaniami z USA i Niemiec, przy czym jest zauważalna wyraźna różnica.

Sesja „Inżynieria oprogramowania"

Zagadnienia automatyzacji procesu przechodzenia od specyfikacji wymagań (w formie tekstu) do schematu konceptualnego zaprezentowali D.R. Knight oraz D.J. Flynn (University of Manchester, Wielka Brytania) w referacie pt. „Automated Conceptual Schema Acquisition". Tekst zawierający wymagania użytkownika może być analizowany z użyciem metod przetwarzania języka naturalnego. Stosowane są reguły heurystyczne oraz analiza semantyczna w celu uzyskania reprezentacji związków encji wybranego problemu. Obecnie jest tworzony prototyp narzędzia mającego realizować to zadanie. Redukuje on rutynowe czynności analityka oraz zwiększa dokładność pracy.

Sesja „Praktyka tworzenia systemów informatycznych"

Charakterystycznym w tej sesji był wygłoszony przez K. Kivistö (University of Oulu, Finlandia) referat „Database Programmer – a need for a new role in object-oriented application development". Referat ten porusza zagadnienia tworzenia zastosowań z wykorzystaniem obiektowych systemów zarządzania bazą danych (OSZBD). Obecnie dostępne OSZBD stwarzają możliwość oprogramowania wybranych metod w samej bazie danych. W związku z tym dla użytkownika powstaje problem, jakie metody powinny być zawarte w bazie danych, a jakie – przechowywane w bibliotece zastosowań. Autor podkreśla wpływ stosowania podejścia obiektowego na szkolenie studentów i profesjonalistów.

Sesja „Narzędzia CASE"

Szczególnie interesujący zestaw referatów z kilku przodujących w świecie ośrodków przedstawiono w tej sesji. Należy tu wymienić referaty:

- o architekturze pakietu KHEOPS: E. Métais i M. Bouzeghoub (Université de Versailles, Francja) – „KHEOPS: a CASE Tool for Object-Oriented Databases",
- o zagadnieniu jakości oprogramowania w świetle użytkowania pakietu CASE: E. M. Gray (University of Strathclyde, Wielka Brytania) – „An Empirical Study of the Critical Factor in Improving the Quality of the Software" oraz I. Rozman, J. Berce, J. Györkös, i B. Stiglic (University of Maribor, Słowenia) – „CASE Tool as a Medium for Indicating Software Unquality"
- o zastosowaniu narzędzi CASE: R. Langerhorst (Cap Gemini Pandata, Utrecht, Holandia) – „User Experiences with SDW".

W ostatnim z w.w. referatów przedstawiono praktyczne doświadczenia przy tworzeniu systemów informatycznych w trzech dużych przedsiębiorstwach w Holandii przy użyciu najpopularniejszego w tym kraju pakietu modułów System Development Workbench (SDW). Pokazano elastyczność stosowania tego pakietu w różnych sytuacjach oraz znaczenie możliwości jego przedefiniowania przez użytkownika, co powoduje rosnącą popularność tego pakietu w Europie Zachodniej.

Sesja „Systemy obiektowo zorientowane"

Szczególne zainteresowanie uczestników konferencji wzbudziła sesja dotycząca systemów obiektowych. Wyniki swoich prac zaprezentowali tu:

- O. Diaz (Universidad del País Vasco, San Sebastian, Hiszpania) i N. Paton (Heriot-Watt University, Edinburgh, Szkocja) – „Extensibility Through Metaclasses in Object-Oriented Systems",
- I. T. Hawryszkiewycz (University of Technology, Sydney, Australia) – „Object Support for Systems Synthesis",
- L. A. Maciaszek, C.N.G. Dampney (Macquarie University, Sydney, Australia) i J.R. Getta (University of Wollongong, Australia) – „From Data Flows to Object Clusters".

W opinii prelegentów i uczestników, systemy obiektowo zorientowane, to jeden z najbardziej aktualnych trendów w dziedzinie współczesnych zastosowań informatyki.

Sesja „Systemy informatyczne – nauczanie i szkolenie"

Jednym z najważniejszych elementów w procesie tworzenia systemów informatycznych (TSI) są zasoby ludzkie. Inwestycje w tym zakresie są traktowane w krajach wysoko rozwiniętych jako przynoszące duże długoterminowe efekty. Zagadnienia edukacji i szkolenia były przedmiotem dwóch referatów: G. Allen i A. R. Jackson (The University of Huddersfield, Wielka Brytania) – „A Pedagogical Information System Development Method" oraz W. Rybiński i S. Wrycza (Uniwersytet Gdański) – „Information Systems Education: Assumptions and Practice". Pierwszy z nich omawiał adaptowanie istniejących metod i narzędzi TSI do potrzeb edukacji. Pokazano, że istniejące metody mają zbyt szeroki zakres działania, podczas gdy zajęcia ze studentami powinny ich kształcić na małych przykładach, a poza tym zajęcia te trwają znacznie krócej niż normalna praca analityka lub projektanta. Autorzy przedstawiają kryteria, jakimi kierowali się dokonując wyboru najbardziej odpowiedniej metody (analiza obiektowo zorientowana). Przeprowadzili ono proces adaptacji tej metody oraz wybrali i wykorzystali narzędzie CASE (IPSYS' KBD).

Drugi referat, oparty na polskich doświadczeniach, opisuje podobną sytuację i zawiera weryfikację propozycji metody nauczania TSI. Efektem jest zestaw ćwiczeń prowadzonych przez dwa semestry, w ciągu których studenci utrwalają wiedzę oraz zdobywają umiejętności użytkowania metod i technik tworzenia systemów informatycznych. W trakcie ćwiczeń studenci pracują nad analizą przypadku w grupach po 4–5 osób, inicjując pracę metodami analizy sytuacyjnej: SWOT, Istotne Czynniki Powodzenia, Sesja Meta-Plan. Do rozwiązania przypadku używają oni metod strukturalnych: Diagramy Przepływu Danych, Słownik Skorowidz Danych, Diagramy Związków Encji i.in. Proces ten jest wspomagany przez pakiet CASE SDW.

Sesja „Tworzenie systemów informatycznych jako praca zespołowa"

Interesujący referat, wskazujący nową dziedzinę zastosowania informatyki

przedstawili D. Ćećez-Kecmanović, O. Kruševac, I. Stanišević i A. Sadiković (University of Sarajevo) – „Representation of Norms in Computer Supported Formal Conversations". Komputerowo wspomagana konwersacja, to nowe medium technologiczne wspomagające kontakty między partnerami w interesach, współpracującymi ze sobą grupami itp. Przedstawiono analizę tych narzędzi, zarówno ze względu na poprawność i zgodność z określonymi normami konwersacji, jak i pod kątem ich efektywności. Opracowano, ilustrowany przykładami, formalny model norm dotyczących takiej konwersacji, oparty na logikach deontycznej i temporalnej.

Sesja „Hipermedia"

Drugim, obok systemów obiektowo zorientowanych, „gorącym" i zgodnym z najnowszymi tendencjami tematem Konferencji, były hipermedia. Ta sesja najbardziej uświadomiła opóźnienie polskiej nauki w stosunku do aktualnego poziomu światowego. Oczywiście badania naukowe są tu uwarunkowane zakupem specjalistycznego sprzętu oraz brakiem kaset i programów w języku polskim. Sesję wypełniły następujące referaty:

- W. Abramowicz, S. Boral, T. Tomaszewski, D. Wlaz (Akademia Ekonomiczna, Poznań) oraz J. Gramowski (University of Vienna, Austria) – „New Prospects in Legal Information Search Hypertext Prototypes",
- C. Boksenbaum, P. Déhais, S. Hammoundi i F. Acosta (Université Montpellier II-CRNS, Francja) „HYPER-AGENDA: a System for Task Management",
- S. Isenmann (Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW) Ulm, RFN) – „Suppporting Argumentative Problem Solving with HYPER_IBIS",
- W. Maes (University of Amsterdam, Holandia) – „Seizing the Strategic Advantage of Multimedia in the Organization",
- D. West (University of Paisley, Szkocja) – "The Process of Knowledge Elicitation for the Design and Development of Multimedia/Hypermedia Decision Suport Aids".

Typowym dla tej sesji był referat S. Isenmanna. Dotyczył systemu HyperIBIS, który pozwala na agregowanie i wyszukiwanie informacji potrzebnych przy podejmowaniu decyzji na wysokim stopniu uogólnienia. Oparty jest on na mechaniźmie „hiper połączeń" i pozwala gromadzić zarówno wycinkową jak i całościową informację znacznie efektywniej niż standardowe systemy informatyczne. System ten opiera się na metodzie IBIS *(Issue Based Information Systems)*.

Sesja „Systemy informatyczne w organizacjach"

Problemy związane z umiejscowieniem tworzenia systemów informatycznych w strukturze organizacji i połączenia tego z zarządzaniem projektem sprawiają często znaczne trudności. Zdarza się, że warunki zewnętrzne zmieniają się w trakcie realizacji projektu co wiąże się z koniecznością korygowania. Zagadnienia te porusza referat K. Enholm (University of Åbo Akademi, Finlandia) i I. Eriksson (University of Colorado, USA) – "Postponed Decisions in Project Management". Przedstawiono w nim badania, których celem jest uzyskanie nowego elastycznego modelu zarządzania projektem. Metoda ta polega na bieżącym reagowaniu na zmieniające się warunki zewnętrzne oraz na stopniowym precyzowaniu końcowego celu projektu, który na początku jest znany tylko z pewnym przybliżeniem.

Sesja „Bazy wiedzy"

Zagadnienia związane ze wspomaganiem procesu twórczego rozwiązywania problemu zawarł J. Pluta (Uniwersytet Gdański) w referacie pt. "Prerequisites to New Theory of Computer Aided Innovative Design. Autor przeprowadził analizę porównawczą metod projektowania oraz analizę obecnego stanu komputerowego wspomagania projektowania innowacyjnego w celu zbudowania komputerowego narzędzia wspomagającego takie projektowanie. Z analizy wynika, że stosunkowo mierne rezultaty tego wspomagania są wynikiem braku jednolitej bazy pojęciowej – teorii, która łączyłaby w zasadniczo nowy sposób najlepsze cechy nowych metod projektowania innowacyjnego z możliwościami sztucznej inteligencji.

Konferencja została zakończona kilkugodzinną, żywą dyskusją panelową na temat "Information Systems Development: Challenges and Chances". Uczestnicy Konferencji potwierdzili m.in. konieczność kontynuowania sopockiego cyklu konferencji. Stwarza on niepowtarzalną możliwość spotkań specjalistów z zakresu metodologii tworzenia systemów informatycznych z Europy Środkowo-Wschodniej oraz krajów o wysokim poziomie technologii informatycznej.

Prof. Stanisław Wrycza
Przewodniczący Międzynarodowego Komitetu Programowego
Mgr Witold Rybiński
Przewodniczący Komitetu Organizacyjnego

1. SYSTEM ROZWOJU ZAWODOWEGO BCS

1.1. Skrócony opis systemu

System Rozwoju Zawodowego (SRZ, ang. *Professional Development Scheme*) Brytyjskiego Towarzystwa Komputerowego (BCS) jest mechanizmem zapewnienia jakości systematycznego szkolenia i rozwoju w dziedzinie informatyki, który pozwala każdemu aktywnemu uczestnikowi programu na osiągnięcie odpowiedniego poziomu wiedzy i doświadczenia na zajmowanym przezeń stanowisku, do którego aspiruje. System został zaplanowany z myślą wspomagania planów rozwoju kwalifikacji kadr stosowanych przez poszczególnych pracodawców, a dla pracodawców, którzy nie posiadają takich planów, SRZ ułatwia ich stworzenie. Ponadto system SRZ stanowi odniesienie indywidualnych planów rozwoju kadr do jednolitych standardów niezależnych od pracodawcy.

Każdy uczestnik jest odpowiedzialny za swój indywidualny program szkoleniowo-rozwojowy obejmujący zarówno plan pracy zawodowej, jak i plan szkolenia zawodowego. Planowanie i realizacja takiego programu przebiegają pod kontrolą opiekuna o odpowiednich kwalifikacjach i kompetencjach, wyznaczonego przez BCS lub przez licencjonowaną organizację uznawaną przez BCS (w Polsce przez PTI). Uczestnik prowadzi specjalny dziennik, w którym odnotowuje przebieg pracy zawodowej i odbyte szkolenia. Postępy w poziomie wiedzy i pracy zawodowej są w określonych odstępach czasu sprawdzane przez opiekuna. Pomyślna realizacja cyklu szkoleniowo-rozwojowego prowadzi do uzyskania nowego stopnia z Modelu Klasyfikacji Zawodowej.

Podstawą systemu jest wyczerpująca tabela wymagań kwalifikacyjnych na poszczególne stanowiska zawodowe, znana jako Model Klasyfikacji Zawodowej (MKZ, ang. *Industry Structure Model*) wprowadzona, sprawdzona i systematy-

* Copyright © by Polskie Towarzystwo Informatyczne, Warszawa 1992

Założenia wdrożenia w RP Systemu Rozwoju Zawodowego na licencji Brytyjskiego Towarzystwa Komputerowego*

cznie udoskonalana przez BCS. Model MKZ definiuje dziesięć poziomów zawodowych w różnych dziedzinach i poddziedzinach zawodowej działalności informatycznej. Zdefiniowane w nim dziedziny i poddziedziny obejmują przeważającą większość współczesnych zastosowań systemów informatycznych.

1.2. Doświadczenia

System Rozwoju Zawodowego jest stosowany w Wielkiej Brytanii od ponad pięciu lat zarówno w sektorze publicznym, jak i prywatnym. Obecnie stosuje go z powodzeniem ponad sto instytucji i przedsiębiorstw. Są wśród nich duże firmy informatyczne: *ICL, AT&T Istel* oraz *Bull HN*, wielcy użytkownicy systemów informatycznych: *British Petroleum, British Telecom, Shell, British Steel*, towarzystwa ubezpieczeniowe *Prudential* i *Royal Insurance*, a także liczne instytucje publiczne, w tym wiele ministerstw. Niektóre firmy międzynarodowe stosują ten system także poza granicami Wielkiej Brytanii.

Według źródeł BCS stosowanie SRZ przynosi następujące korzyści dla przystępujących do niego organizacji:

– przyciągnięcie lepszych pracowników i zmniejszenie migracji kadr,
– możliwość lepszego planowania,
– lepszy rozdział zasobów kadrowych,
– poprawienie wydajności pracy informatyków, zwiększenia ich fachowości, podniesienie prestiżu firmy na zewnątrz, zwiększenie zaufania do przemysłu informatycznego w ogóle,
– wdrożenie norm, które niedługo staną się normami EWG.

Korzyści dla pracowników przystępujących do SRZ są według tych samych źródeł następujące:

– możliwość aktywnego planowania swej kariery zawodowej, wpływu na awanse i ubiegania się o stopnie zawodowe,
– trwałe, potwierdzone świadectwa dorobku zawodowego,
– pisemne i powszechnie uznawane potwierdzenie własnych kwalifikacji zawodowych.

Niedawno licencję na system SRZ na obszar USA otrzymało stowarzyszenie *Data Processing Managers Association*, co z dużym prawdopodobieństwem rozszerzy stosowanie tego systemu w Stanach Zjednoczonych. W trakcie finalizacji jest licencja SRZ na obszar Singapuru.

Model Klasyfikacji Zawodowej, leżący u źródeł systemu SRZ, stał się wzorem dla programu przygotowania Europejskiej Klasyfikacji Kwalifikacji Informatycznych (EKKI) podjętego przez ogólnoeuropejską grupę roboczą stworzoną przez *Council of European Professional Informatics Societies* (CEPIS) z poparciem i pomocą finansową Komisji EWG. W pracach tej grupy znaczący udział bierze Polskie Towarzystwo Informatyczne.

Założenia programu EKKI przewidują stworzenie wspólnego zestawu wymagań zawodowych dla informatyków, który znajdzie następujące zastosowania:
– będzie stanowił międzynarodowy „wspólny język" wspomagający łatwiejsze przemieszczanie się wyspecjalizowanych fachowców w dziedzinie informatyki,
– będzie sprzyjał większej spójności terminologii w Europie,
– wspomoże wdrożenie Dyrektywy EWG związanej z wzajemnym uznawaniem kwalifikacji zawodowych.

Obecnie prace nad programem EKKI weszły w fazę wersji narodowych, które uwzględniają lokalne systemy edukacyjne. W Polsce takie prace prowadzi PTI.

2. Korzyści z zastosowania systemu SRZ w Polsce

Zastosowania informatyki stanowią jeden z podstawowych elementów w przekształceniu ustroju społeczno-gospodarczego Polski. Znacząca rola tych zastosowań została szczegółowo omówiona w *Propozycji strategii rozwoju informatyki i jej zastosowań w Rzeczypospolitej Polskiej*, przygotowanym przez PTI na zlecenie Rady Ministrów RP w lipcu 1991 r. Wdrożenie Systemu Rozwoju Zawodowego wspomoże takie przekształcenia przez:
2.1. Wprowadzenie wypróbowanych rozwiązań związanych ze szkoleniem i rozwojem informatyków, opartych na rozwiązaniach zgodnych z zaleceniami EWG.
2.2. Dostarczenie polskim przedsiębiorstwom i instytucjom zunifikowanych metod oceny pracy zawodowej i kryteriów awansu informatyków.
2.3. Wprowadzenie zgodnego z praktykami europejskimi systematycznego podejścia do spraw kwalifikacji zawodowych, który może znaleźć zastosowanie także w innych dziedzinach działalności fachowej.
2.4. Uzyskanie fachowego doradztwa i pomocy dzięki nadzorowi BCS nad szczegółowym wdrożeniem systemu SRZ w Polsce.
2.5. Umożliwienie wybranym przedstawicielom PTI i organizacji stosujących SRZ uzyskanie praktycznych doświadczeń z zastosowania SRZ w Wielkiej Brytanii w wielu różnych środowiskach.
2.6. Umożliwienie polskim informatykom udziału w pracach Komisji Zarządzania SRZ w Wielkiej Brytanii i przez to wpływu na rozwój systemu SRZ.
2.7. Doradztwo i pomoc BCS w praktycznym wdrażaniu i zarządzaniu systemem SRZ.
2.8. Rozbudzenie i zaspokojenie ambicji zawodowych polskich informatyków pozbawionych ostatnio formalnych możliwości rozwoju i potwierdzenia swych kwalifikacji w wyniku upadku systemu stopni zawodowych przyznawanych dawniej przez NOT, zresztą na podstawie niejasnych kryteriów.

3. Najważniejsze postanowienia projektu umowy licencyjnej SRZ

3.1. Zobowiązania PTI

3.1.1. PTI znajdzie i mianuje Zarządcę projektu SRZ. Zarządca będzie doświadczonym informatykiem zawodowym, ściśle związanym z kształceniem i rozwojem zawodowym informatyków. Zarządca powinien płynnie posługiwać się językiem angielskim, ponieważ przejdzie począt-

kowe szkolenie w Wielkiej Brytanii, zorganizowane przez BCS.
3.1.2. PTI sfinansuje i wykona wszystkie niezbędne przekłady dokumentacji w języku angielskim na język polski, a także reprodukcję dokumentacji w języku polskim.
3.1.3. PTI z pomocą BCS starannie wybierze niewielką grupę polskich organizacji (3–6), które będą stanowić grupę pilotującą wdrożenie SRZ w Polsce. Wyselekcjonowana grupa organizacji stworzy zaczątek pierwszej Komisji Zarządzania SRZ w Polsce.
3.1.4. PTI zgodnie ze wskazówkami BCS upoważni „opiekunów" mianowanych przez użytkowników systemu do sprawowania koordynacji nad wdrażaniem systemu w ich organizacjach. Opiekunowie powinni mieć odpowiednie kwalifikacje i doświadczenie do sprawowania swych funkcji.
3.1.5. PTI umocni powiązania z administracją państwową, uczelniami wyższymi, związkami pracodawców i związkami zawodowymi w celu zapewnienia ich zrozumienia i poparcia dla systemu SRZ, potwierdzając w ten sposób współdziałanie polskiego środowiska zawodowego we wdrażaniu systemu.
3.1.6. PTI stworzy odpowiednią strategię marketingową i reklamową w celu zapewnienia dobrego klimatu poparcia dla początkowego wdrożenia i późniejszego rozwoju systemu SRZ.
3.1.7. PTI stworzy odpowiednio wyposażoną w środki finansowe komórkę administracyjną podtrzymującą działania systemu, jego operacje finansowe oraz kontakty z użytkownikami systemu i BCS.

3.2. Zobowiązania BCS

3.2.1. BCS udostępni PTI w języku angielskim wszystkie zaktualizowane materiały dotyczące systemu SRZ, w tym poradniki zarządzania systemem i inne materiały pisemne związane z SRZ. Krótkie wprowadzenie do systemu SRZ zostanie udostępnione w języku polskim.
3.2.2. BCS zorganizuje wstępne i następne pobyty zapoznawcze u użytkowników SRD w Wielkiej Brytanii dla polskich uczestników SRZ wybranych przez PTI. Część czasu takich pobytów zostanie poświęcona na udział w sesjach szkoleniowych zorganizowanych przez lub na rzecz BCS.
3.2.3. W miarę możliwości okres pobytów zapoznawczych będzie zbieżny z zaplanowanymi posiedzeniami Komisji Zarządzania SRZ.
3.2.4. BCS wybierze dwóch doświadczonych konsultantów systemu SRZ, którzy będą stale współpracować z PTI i będą źródłem fachowego doradztwa podczas początkowej fazy wdrażania SRZ, a później będą służyć radą w miarę potrzeb.
3.2.5.BCS pomoże i doradzi PTI w sprawach wymagań administracyjnych związanych ze sprawnym zarządzaniem i kontrolą finansową podczas wdrażania SRZ.

4. Harmonogram wdrażania SRZ

4.1. Rok pierwszy

4.1.1. Wyznaczenie Zarządcy projektu SRZ przez PTI. Wytypowanie organizacji (3–6) do udziału w programie pilotowym. Wyznaczenie opiekunów PTI dla poszczególnych organizacji pilotowych. Zorganizowanie biura administrującego systemem SRZ.
4.1.2. Zapoznawczy tygodniowy pobyt w Wielkiej Brytanii Zarządcy projektu SRZ oraz dwóch tygodni innych delegatów, po jednym z instytucji publicznej i przedsiębiorstwa.
4.1.3. Tygodniowy pobyt w Polsce dwóch przedstawicieli BCS w celu pomocy w zaplanowaniu szczegółowego programu wdrażania systemu, ustalenia szczegółów jego realizacji oraz wizyt w organizacjach pilotowych.
4.1.4. Przekład dokumentacji związanej z SRZ i jej przystosowanie do polskich realiów.
Uwaga: Ten punkt harmonogramu będzie skoordynowany z pracami polskiej grupy roboczej PTI nad przekładem i przystosowaniem Europejskiej Klasyfikacji Kwalifikacji Informatycznych.
4.1.5. Tygodniowy pobyt w Wielkiej Brytanii koordynatorów SRZ z organizacji pilotowych poświęcony szkoleniu początkowemu i wizytom w organizacjach brytyjskich stosujących SRZ.
4.1.6. Ustalenie początkowych poziomów zawodowych dla uczestników programu w organizacjach pilotowych. Wdrożenie programu rozwoju zawodowego i kształcenia w organizacjach pilotowych.
4.1.7. Tygodniowy pobyt w Polsce dwóch konsultantów BCS poświęcony przeglądowi wstępnych wyników wdrożenia systemu SRZ i rozwiązaniu problemów związanych z tym wdrożeniem.

4.2. Rok drugi

4.2.1. Kontynuacja wdrażania systemu SRZ w organizacjach pilotowych.
4.2.2. Rozpoczęcie wdrażania SRZ w następnych organizacjach.
Uwagi: Jeśli pilotowe wdrażanie będzie przebiegać pomyślnie, przewidywane jest dołączenie nowych organizacji do programu już w pierwszym roku.
4.2.3. Tygodniowy lub dwutygodniowy pobyt w Wielkiej Brytanii koordynatorów SRZ z organizacji pilotowych poświęcony szkoleniu uzupełniającemu i wizytom w organizacjach brytyjskich stosujących SRZ w celu porównania doświadczeń, a także udziału w Komisji Zarządzającej SRZ.
4.2.4. Tygodniowy pobyt w Polsce dwóch konsultantów BCS w celu sprawdzenia postępów wdrażania i doradztwa dla organizacji pilotowych i innych organizacji, które przyłączyły się do systemu.
4.2.5. Współpraca PTI z BCS nad dalszym wdrażaniem systemu.

5. Uwagi dla organizacji zainteresowanych programem

Przedsiębiorstwa i instytucje zainteresowane udziałem w programie pilotowym wdrożenia SRZ są proszone o skontaktowanie się z jednym z dwóch koordynatorów prac wstępnych z ramienia PTI:

Piotr Fuglewicz
tel. sł. (03) 154-22-65 lub
(032) 59-60-51 (prosić CSBI)
dom (032) 51-05-79
Paweł Gizbert-Studnicki
tel sł. (012) 34-03-39 lub
(012) 34-28-63
fax (012) 33-36-24
dom (012) 33-42-71

6. Źródła

W niniejszym opracowaniu wykorzystano następujące źródła:

1. *Proposal,* British Computer Society, April 14, 1992
2. *Industry Structure Model* (Release 2), BCS, 1991
3. *Professional Development Scheme, Management guide,* BCS, 1991
4. *Professional Development Scheme, Patricipants guide,* BCS 1991
5. W. M. Turski, *O brytyjskich stopniach zawodowych,* Biuletyn PTI, luty 1991

Opracowanie:
Piotr Fuglewicz i Paweł Gizbert-Studnicki
Polskie Towarzystwo Informatyczne
Warszawa, maj 1992

Przedruk z Biuletynu PTI nr 9–10, 1992 r.

Szybki rozwój systemów Windows

Gdy przed laty firma Microsoft zaczęła oferować pierwsze wersje graficznego sprzętu użytkownika (*Graphical User Interface* – GUI) o nazwie Windows, prawie nikt nie przewidywał jego zawrotnej kariery. Jeszcze w 1989 r. krytykowano te systemy jako nieporęczne i wolne w działaniu, a IBM i inni producenci traktowali je jako stopień pośredni do systemu operacyjnego OS/2. Niemal wszyscy byli pewni, że na rynku oprogramowania będzie się umacniała dominacja wielkich firm opierających się na DOS-ie. Po pierwszych sukcesach zaanonsowanego w marcu 1990 r. systemu Windows 3.0, w połowie 1991 r. nikt już nie miał wątpliwości, że było to istotne wydarzenie. Pojawienie się w kwietniu 1992 r. znacznie usprawnionej wersji 3.1 spowodowało znaczny wzrost aplikacji pracujących pod tym systemem, obejmując tak popularne programy jak Excel, Word i Power-Point, które odebrały znaczne udziały rynku innym firmom. Obecnie jest przygotowywana wersja NT (New Technology), będąca już 32-bitowym systemem operacyjnym, który stanowić będzie poważną konkurencją dla systemu IBM OS/2 2.0. Inne narzędzia firmy Microsoft z dziedziny baz danych, takie jak Access, Fox-Pro i SQL Server stanowią zagrożenie dla produktów takich firm, jak Borland i Oracle, a możliwości sieciowe wersji NT podwajają możliwość współzawodnictwa firm Microsoft i Novell. Prawie wszystkie firmy wytwarzające oprogramowanie wykorzystują obecnie system Windows. Według oceny firmy Software Publishers Association w pierwszej połowie 1992 r. rynek oprogramowania Windows osiągnął obroty w wysokości 857 mln dolarów, co oznacza wzrost o ponad 141% w porównaniu do pierwszej połowy 1991 r. W tym samym czasie wartość programów opracowanych pod DOS-em zmniejszyła się o 8,7% do wartości 1,3 mld dolarów, a wartość programów dla komputerów Macintosch firmy *Apple* wzrosła o 32% do poziomu 490 mln dolarów. Przewidywano, że już w trzecim kwartale 1992 r. wartość sprzedaży oprogramowania Windows przekroczy wartość oprogramowania DOS.

Wpływ rozwoju oprogramowania Windows na produkowany sprzęt komputerów osobistych był również znaczny. Jeszcze, niedawno dominowały tu systemy z mikroprocesorami 286, pamięcią operacyjną 1 MB i stacjami dyskowymi o pojemności 20 MB. Obecnie wymagane są urządzenia z mikroprocesorem co najmniej 386, pamięcią operacyjną 4 MB i stacją dyskową od 60 MB wzwyż. Oczekuje się, że połowa systemów wyprodukowanych w 1992 r. jest dostosowana już do oprogramowania Windows. Coraz więcej systemów jest wyposażonych w techniki towarzyszące (np. płyty przyspieszania operacji graficznych i układ wizyjny z szyną lokalną), które pomagają spełniać wymagania dla pracy z oprogramowaniem Windows.

Wraz z rozwojem 32-bitowych systemów operacyjnych, takich jak OS/2 i Windows NT, rozwój sprzętu skoncentruje się na procesach 32-bitowych. Wg przeprowadzonej przez firmę Dataquest analizy dotyczącej już zainstalowanych i przewidzianej produkcji komputerów osobistych do 1996 r. wynika, że nie przewiduje się rozwoju maszyn opartych na mikroprocesorach 286, a tylko niewielki rozwój systemów 8086 przewidziano dla komputerów kieszonkowych. Najwięcej zostanie wyprodukowanych systemów opartych na mikroprocesorach 486 SX/SL i właśnie one zyskują najwięcej zamówień. Drugie miejsce zajmują układy oparte na procesorach 386 SX/SL, natomiast trzecie – zainstalowane i jeszcze eksploatowane stare komputery oparte na mikroprocesorach 8088/8086, a wśród perspektywicznych – 486DX. Według tych danych, nowy układ firmy Intel o symbolu P5 lub 586 nie będzie wówczas jeszcze stanowił znaczącego udziału wśród zainstalowanych i zamawianych komputerów.

Z przewidywanego w najbliższych latach przez firmę Disk/Trend rozwoju stacji dyskowych wynika, że od 1993 r. nie będą już wytwarzane stacje o pojemności poniżej 30 MB, a w 1994 r. ponad 30% stacji będzie miało pojemność 100–300 MB, a ok. 70% – 60–300 MB.

System Windows na tym polu nie jest jednak jedynym. Wiadomo, że od dawna działania w tym kierunku podejmowano w odniesieniu do komputerów Macintosch firmy Apple. Słabą stroną jest tu jednak to, że system operacyjny Mac OS działa wyłącznie na tych komputerach. Może to jednak ulec zmianie po opracowaniu na podstawie umowy IBM-Apple systemu PowerOpen. Natomiast Microsoft przewiduje implementację wersji NT na układy firm Intel i Mips oraz procesory Alpha firmy DEC. W ten sposób oprogramowanie Windows może stać się uniwersalnym. Opracowując wersję 3.0 firma Microsoft nie planowała tak szerokiego zasięgu, ale rozwój wydarzeń spowodował, że został określony charakter przyszłych systemów operacyjnych. Okazało się, że sprzęt użytkownika i programowania pozostają w różnych dziedzinach stałe, natomiast jądro systemu operacyjnego sprawi, że sprzęt będzie niewidoczny dla użytkownika.

Jesienią 1992 r. sprzedano ponad 10 mln kopii oprogramowania Windows. Sama firma Microsoft przyznaje, że prawdopodobnie tylko 30% tych zakupów jest w pełni użytkowane. Oznacza to, że na razie baza użytkowników Windows jest jeszcze dwukrotnie mniejsza, od bazy użytkowników systemu Macintosha i ponad dziesięciokrotnie mniejsza od bazy użytkowników DOS. Przyszłość jednak rysuje się dla Windows bardzo pomyślnie i analitycy przewidują, że 1997 r. obie wersje Windows, a więc działające pod DOS-em oraz NT, ilościowo będą stanowić 83% sprzedanych systemów, podczas, gdy system UNIX – mniej niż 2%. Inne prognozy, oparte na ankiecie miesięcznika Byte, określają do tego samego roku 47% udziału dla DOS-Windows, 17% dla NT, 16% dla OS/2 oraz 7% dla UNIX-a. Inną miarą jest oczekiwana dla poszczególnych systemów liczba aplikacji. Wg firmy Forrester Research liczba zastosowań Windows podwoi się w okresie od jesieni 1992 r. do 1995 r. z 5000 do 10 000. Aplikacje OS/2 będą rosły szybciej, ale w 1995 r. osiągnął tylko 3000, a programy na Macintosha wzrosną z 4000 do 5000. Spadek cen oprogramowania nie będzie jednak tak duży jak sprzętu. Aby wyeliminować konkurentów, firma Microsoft obniżyła znacznie ceny, zwłaszcza dla znaczących klientów. Oznacza to, że dostosowane do Windows programy Excel lub Word można nabyć za cenę poniżej 100 dolarów w przypadku zakupu całego zestawu Windows. Wg czasopisma „Soft Letter", w przyszłości ceny będą musiały prawdopodobnie wzrosnąć, aby osiągnąć zadowalający zysk. Firma Dataquest natomiast oceniła spadek cen oprogramowania Windows na 6% w półroczu obejmującym II i III kwartał 1992 r. oraz

przewiduje 10% spadek w 1993 r. Dotyczy to jednak sprzedaży detalicznej.

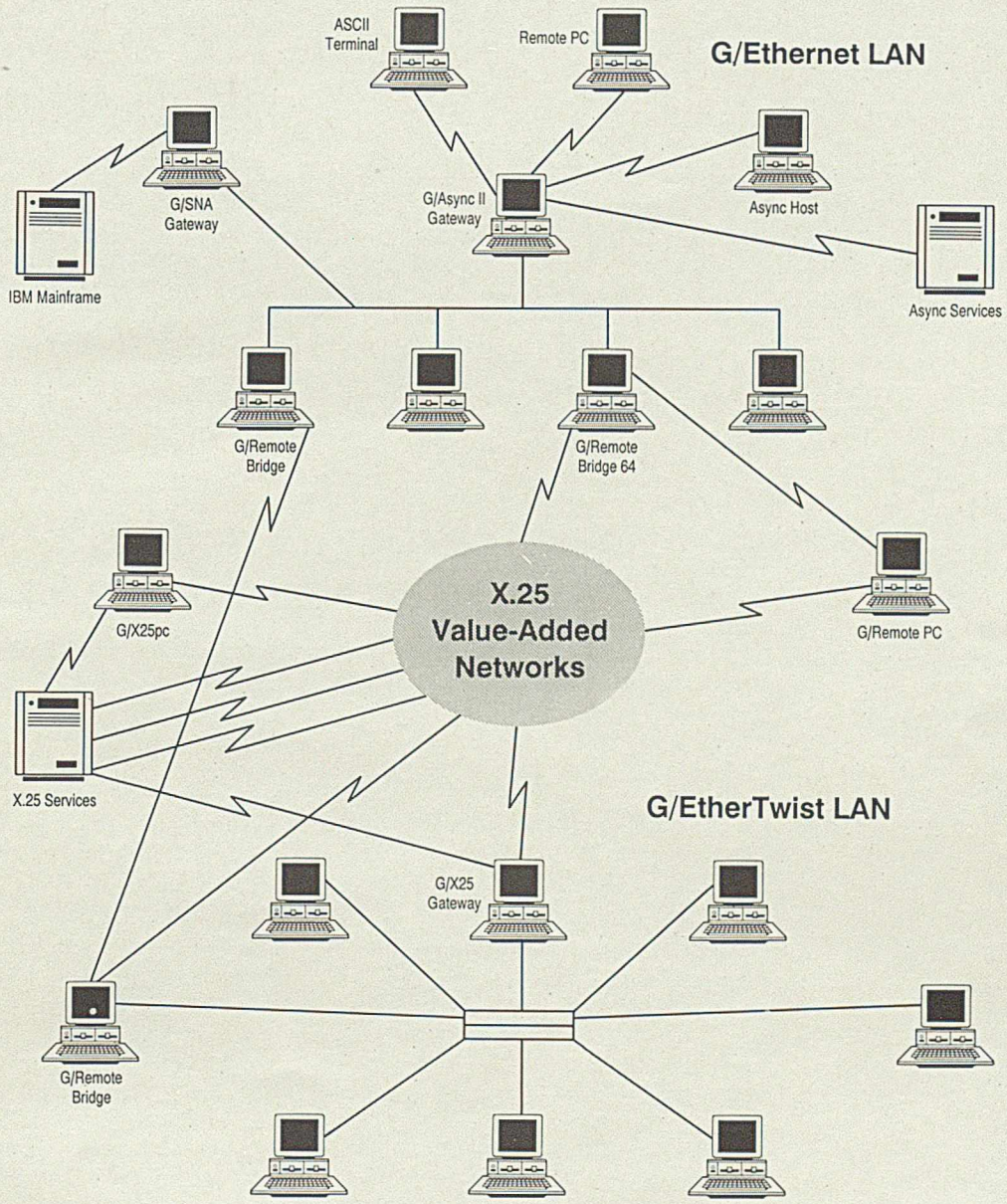
Pojawienie się wersji NT będzie znaczącym wydarzeniem dla 32-bitowych systemów operacyjnych, ale w połowie lat dziewięćdziesiątych pojawią się systemy ukierunkowane na obiekty (ang. *object-oriented*), takie jak IBM/Apple Taligent czy Sun Spring. Aby podjąć to wyzwanie, Microsoft przygotowuje na 1994 r. wersję Windows o nazwie Cairo. Będzie ona oparta na istniejących jądrach systemów DOS i NT, co według projektantów ma zapewnić ciągłość w przechodzeniu od istniejących dziś programów. Wersja ta opiera się na istniejącym już protokole OLE i rozszerza jego zastosowania sieciowe.

W związku z przewidywanym szybkim rozwojem komputerów podręcznych (*pocket-size* i *plam-top*), firma Microsoft podjęła działania również na tym polu tworząc wersję Windows 3.1 umieszczoną w pamięci ROM, oraz opracowując Windows for Pen Computing, a więc oprogramowanie przystosowane do małych komputerów z użyciem pióra. **J.R.**

Czajkowski G., Zieliński K.: Linda – środowisko do przetwarzania równoległego i rozproszonego w sieciach stacji roboczych INFORMATYKA 1993, nr 5, s. 1 Charakterystyka narzędzia programowego Linda firmy Scientific Computing Associates, ułatwiającego koordynację programów napisanych w językach standardowych oraz omówienie i przykłady stosowania implementacji Lindy dla języka C.	**Czajkowski G., Zieliński K.: Linda – an environment for parallel and distributed processing in workstation networks** INFORMATYKA 1993, No. 5, p. 1 Characteristics of the Linda programming tool, a product of Scientific Computing Associates, which simplyfies coordination of programs written in standard languages, as well as discussion and examples of the Linda implementation for the C language.	**Czajkowski G., Zieliński K.: Linda – eine Umwelt für parallele und zerstreute EDV in Workstation-Netzwerken** INFORMATYKA 1993, Nr. 5 , S. 1 Eine Charakteristik des Linda-Programmhilfsmittels, eines Produktes der Firma Scientific Computing Associates, das die Koordination der in standarisierten Sprachen geschriebenen Programme erleichtert, sowie eine Besprechung und Anwendungsbeispiele der Linda-Implementierung für die C-Sprache.
Bartoszek D.: ECHO – komputerowy serwis informacyjny INFORMATYKA 1993, nr 5, s. 6 Charakterystyka zawartości oraz możliwości korzystania z systemu ECHO, jednego z komputerowych serwisów informacyjnych Europejskiej Wspólnoty Gospodarczej.	**Bartoszek D.: ECHO – the computerized information service** INFORMATYKA 1993, No 5, p. 6 Characteristics of content and of using possibilities of the ECHO system, one of the EEC's computerized information services.	**Bartoszek D.: ECHO – ein computerisierter Informationsservice** INFORMATYKA 1993, Nr. 5, S. 6 Eine Charakteristik von Inhalt und Ausnutzungsmöglichkeiten des ECHO-Systems, eines von den computerisierten EWG-Informationsservicen.
Kotzian L.: Prosta integracja środowiska sieci NetWare ze światem UNIX INFORMATYKA 1993, nr 5, s. 14 Charakterystyka produktu programowego LAN Workplace for DOS firmy Novell, ułatwiającego integrację środowiska sieci NetWare ze środowiskiem UNIX.	**Kotzian L.: Simple integration of the NetWare network environment with the UNIX world** INFORMATYKA 1993, No 5, p. 14 Charakteristics of the Novell's LAN Workplace for DOS program product, which simplyfies integration of the NetWare network environment with the UNIX environment.	**Kotzian L.: Eine einfache Integration von Umwelt des NetWare-Netzes mit der UNIX-Welt** INFORMATYKA 1993, Nr. 5, S. 14 Eine Charakteristik des LAN Workplace for DOS-Programmproduktes von Firma Novell, das eine Integration der Umwelt von NetWare-Netz mit der UNIX-Umwelt erleichtert.
Gaik D., Przybylski A.: Rekomendacje narzędzi graficznych dla środowiska APW INFORMATYKA 1993, nr 5, s. 17 Możliwości włączenia do pakietu APW zaawansowanych edytorów graficznych, rozszerzających w nim zakres prezentacji graficznej.	**Gaik D., Przybylski A.: Graphic tool recomendation for the APW environment** INFORMATYKA 1993, No. 5, p.17 Possibility of advanced graphic editors inclusion to the APW program package for scope increase of graphic presentation.	**Gaik D., Przybylski A.: Empfehlungen der graphischen Hilfsmittel für die APW-Umwelt** INFORMATYKA 1993, Nr. 5, S. 17 Möglichkeiten der Einbeziehung in das APW-Paket von fortgeschrittenen Graphikeditoren, die den Bereich der graphischen Präsentation erweitern.
Goliński J.: Problemy projektowania systemów informacyjnych INFORMATYKA 1993, nr 5, s. 19 Charakterystyka ewolucji metod projektowania systemów informacyjnych oraz porównanie mitów związanych ze zjawiskiem tzw. kryzysu softwarowego z konkretną rzeczywistością.	**Goliński J.: Problems of information systems designing** INFORMATYKA 1993, No. 5, p. 19 Characteristics of the evolution of information systems designing methods and comparison of myths connected with so-called „software crisis" to concrete reality.	**Goliński J.: Probleme der Informationssystemeprojektierung** INFORMATYKA 1993, Nr. 5, S. 19 Eine Charakteristik der Evolution von Methoden der Informationssystemeprojektierung und ein Vergleich von Mythen der sogenannten „Software-Krise" mit konkreter Realität.

Produkty sieciowe LAN i WAN

Gateway communications, inc.



BEZKONKURENCYJNE PRODUKTY
SIECI KOMPUTEROWYCH NAGRADZANE PRZEZ:

PC MAGAZINE, LAN MAGAZINE, INFO WORLD

OFERUJE AUTORYZOWANY DYSTRYBUTOR:

MIKROB

PC MAGAZINE EDITORS' CHOICE January 31, 1989 G/Ethernet AT

PC MAGAZINE EDITORS' CHOICE October 16, 1990 G/EtherTwist

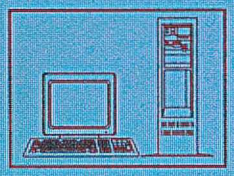
P.W.P.T. MIKROB SP. Z O.O.

20-346 Lublin, ul. Długa 5
Tel. (0-81) 420-61, faks 415-43, teleks 643776 MIKRO PL

0/15/91

Na życzenie wysyłamy katalog produktów. Atrakcyjny program współpracy dla dealer'ów.

JUNISOFTEX®



Przedsiębiorstwo Produkcyjno-Usługowe Innowacji Technicznych Biuro Techniczno-Handlowe 44-100 Gliwice ul. Konstytucj 11.
tel. 31-57-00, 31-51-31, 31-75-10, fax 31-75-10, centr. 31-90-81 do 88, telex 036233
Zespół wdrożeń i analiz systemów komputerowych wewn. 15 ,25, 282, Zespół inżynierii oprogramowania wewn. 14, 20, 245,
Zespół d/s serwisu systemów Novell i sprzętu komputerowego wewn. 16, 245, Dział Handlowy wewn. 18, 28, 201,
Szkoła Komputerowa wewn. 282, Sekretariat wewn. 250, Prezes wewn. 13, 272, Dyrektor wewn. 12, 272, Główny Księgowy wewn. 256,
** Konto: PKO I/O Gliwice 27661-225298-136 **

OFERUJEMY

Kompleksową komputeryzację przedsiębiorstw obejmująca dostawę i wdrożenie zintegrowanych systemów komputerowych takich jak system:

finansowo-kosztowy
(automatyczne rozliczanie kosztów)
+
gospodarka materiałowa
+
środki trwałe
+
wyposażenie
+
obrót towarowy fakturowanie sprzedaży
+
kadry-płace
+
techniczne przygotowanie produkcji

Sprzęt komputerowy w tym:

Komputery
Monitory
Dyski twarde
Drukarki
(mozaikowe i laserowe)
osprzęt sieciowy
(karty sieciowe, kable zwykłe i światłowodowe)

Systemy i oprogramowanie licencjonowane

DOS
Novell
Btrieve
SQL Base
Windows

Szkolenia komputerowe

podstawy obsługi komputerów
arkusze kalkulacyjne (Qpro, Lotus, EXCEL)
edytory tekstów (WordPerfect, TAG, ChiWriter)
bazy danych (dbase IV, Clipper, Paradox)
systemy operacyjne (DOS, Novell)

SOFTWARE SUPPORT FOR DOS, NOVELL 2.2, 3.11, 4.0, SQL BASE firmy GUPTA

*

OPROGRAMOWANIE DLA PRZEDSIĘBIORSTW

oferujemy własnej produkcji zakładowy system informatyczny PHU-Perfect(N) obejmujący swoim zakresem:
system finansowo-kosztowy umożliwiający automatyczne rozliczanie kosztów, atomatyczną dekretację, analizę rozrachunków
system obrotu materiałami, wyrobami, towarami (kody paskowe) w tym: stany magazynowe, ewidencje zamówień, fakturowanie sprzedaży (VAT) z automatyczną dekretacją rozdzielników kosztów i sprzedaży do systemu finansowo-kosztowego,
system ewidencji majątku trwałego i wyposażenia
system kadrowo-płacowy (SQL) z automatyczną dekretacją rozdzielników płacowych do systemu finansowo-kosztowego
system technicznego przygotowania produkcji (uk. 1993 r)
(oprogramowanie w systemie SQL BASE w przygotowaniu)

*

OPROGRAMOWANIE DLA URZĘDÓW MIAST I GMIN

oferujemy własnej produkcji system finansowo-podatkowy UMG-Perfect(N) obejmujący swoim zakresem:
system finansowo-księgowy,
systemy podatkowe obejmujące: ustalanie wymiarów, naliczanie i kontrolę płatności należności podatkowych dla dowolnych podatników i/lub płatników podatków na ich kontach i kontach dochodów ze swobodną korektą dowolnych informacji w dowolnym okresie a w tym:
ewidencja zapłat z wykorzystaniem kodów paskowych, emisja wezwań do zapłaty, naliczanie odsetek.
system ewidencji majątku

*

Zapraszamy do współpracy w zakresie wdrożen naszych systemów partnerów, - firmy informatyczne z całego kraju !!!

0/7/92