

P. 1877/92

3

1992

informatyka

RD

Miesięcznik
ISSN 0542-9951
INDEKS 36124

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
mgr inż. Piotr FUGLEWICZ
mgr Teresa JABŁOŃSKA
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
dr inż. Wojciech MOKRZYCKI
mgr inż. Jan RYZKO
dr Zdzisław SZYJEWSKI
mgr Hanna WŁODARSKA –
– MARCZENKO

**PRZEWODNICZĄCY
RADY PROGRAMOWEJ:**

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:

Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT
Spółka z o.o.
ul. Biała 4
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. Biała 4
telefon: 20-31-24
telefaks: 20-31-16
teleks: 814550**

W numerze:

	Strona
Propozycja strategii rozwoju informatyki i jej zastosowań w Rzeczypospolitej Polskiej (3)	1
Nowe techniki w konstruowaniu interfejsów użytkownika – <i>Mariusz Borsa</i>	12
Od BTRIEVE-a do SQL-a. Nowoczesne, wielopoziomowe środowisko baz danych firmy NOVELL – <i>Jan Baranowski</i>	15
Zabawa w słówka, czyli kilka informacji o językoznawstwie statystycznym – <i>Piotr Fuglewicz</i>	20
Strukturalna analiza i projektowanie systemów informatycznych – metodyka Gane'a-Sarsona – <i>Katarzyna Stapor</i>	25
<i>/unix</i>	
Integracja systemów operacyjnych MS-DOS i UNIX (3) – <i>Paweł Żak</i>	27
Ze świata	
Perspektywy systemów 64-bitowych – <i>Jan Ryżko</i>	30
Z kraju	
II krajowa konferencja metodyki nauczania przedmiotów informatycznych w uczelniach rolniczych – <i>Jadwiga Orylska</i>	III okł.

W najbliższych numerach:

- Maria Jerczyńska prezentuje swoje poglądy na temat komputeryzacji w obrocie towarowym.
- Krystyna Koleśnik opisuje próbę zastosowania metodologii obiektowej, a także programowania obiektowego, do programowania symulacji z wykorzystaniem języka Turbo Pascal 5.5.
- Iwona Dąbrowska i Dorota Depowska zajmują się metodą automatycznego rozpoznawania znaków.
- Wiesław Przetacznik charakteryzuje graficzne stacje robocze SPARCstation.
- Kazimierz Kowalski i Andrzej Siemiński omawiają zastosowanie sieci komputerowych w procesie dydaktycznym i przedstawiają możliwości jakie uzyskuje użytkownik przez komunikowanie się komputerowo.

Warunki prenumeraty

Przyjęcie prenumeraty – wyłącznie na podstawie dokonanej wpłaty na drukach dostarczanych dotychczasowym prenumeratorem przez Wydawnictwo, lub nowym – po uprzednim zgłoszeniu zapotrzebowania (pisemnie lub telefonicznie) w Zakładzie Kolportażu Wydawnictwa.

Blankiet wpłaty – powinien zawierać następujące informacje: dokładna nazwa i adres (z kodem pocztowym) zamawiającego, tytuły zamawianych czasopism, ich liczbę i okres prenumeraty.

Wpłata – zgodnie z podanymi cenami należy dokonać w banku lub w UPT na konto podane na naszym blankiecie, tj:

Państwowy Bank Kredytowy III O/Warszawa nr: 370015-1573-139-11

Prenumeratory zbiorowi – osoby prawne obowiązują blankiety „Wpłata-Zamówienie”. Cena normalna.

Prenumeratory indywidualni – osoby fizyczne obowiązują blankiety typu przekazy dla wpłat na rachunki bankowe. Cena normalna.

Prenumerata ulgowa – zgodnie z podaną ceną ulgową przysługuje wyłącznie osobom fizycznym, będącym członkami SNT, studentom i uczniom szkół zawodowych. Uczniowie szkół ogólnokształcących mogą zamówić w prenumeracie ulgowej tylko miesięcznik „Aura”

Uwaga! w podanym okresie prenumeraty można zamówić tylko po jednym egzemplarzu z każdego tytułu.

Prenumerata ze zleceniem wysyłki za granicę – cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa od ceny normalnej.

Należy podać dokładny adres odbiorcy za granicą.

Terminy przyjmowania prenumeraty:

- do 10 listopada na I, II, III, IV kwartał następnego roku
- do 28 lutego na II, III, IV kwartał br.
- do 31 maja na III i IV kwartał br.
- do 31 sierpnia na IV kwartał br.

Zmiany w prenumeracie, np. zmiana liczby tytułów, liczby egzemplarzy, rezygnacja z prenumeraty, można zgłaszać tylko w podanych terminach z mocą obowiązującą od następnego kwartału.

Egzemplarze archiwalne (z lat ubiegłych)

Można nabyć za gotówkę w Klubie Prasy Technicznej, Warszawa, ul. Mazowiecka 12 (tel. 26-80-16) lub zamówić pisemnie w Zakładzie Kolportażu, Dział Handlowy, 00-950 Warszawa, skr. poczt. 1004 (tel. 40-37-31), na rachunek lub za zaliczeniem pocztowym.

Informacji o prenumeracie udziela: Zakład Kolportażu Wydawnictwa SIGMA-NOT Spółka z o.o., 00-716 Warszawa, ul. Bartycycka 20, skr. 1004. Telefony: 40-00-21 wewn. 293, 295, 299 lub 40-30-86, 40-35-89.

Wstępna cena jednego egzemplarza na 1991 rok: normalna – 15000 zł ulgowa – 11250zł

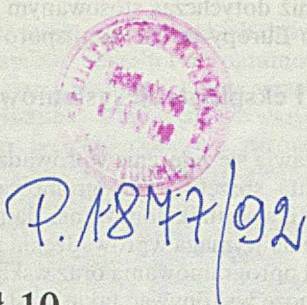
Wartość prenumeraty (w zł):

Normalna: kwartalna – 45000, półroczna 90000, roczna 180000

Ulgowa: kwartalna – 33750, półroczna 67500, roczna 135000

Uwaga: W przypadku zmiany cen w okresie objętym prenumeratą, prenumeratory zobowiązani są do dopłaty różnicy cen.

Propozycja strategii rozwoju informatyki i jej zastosowań w Rzeczypospolitej Polskiej (3)



Rozdział 10

Standaryzacja

Standardy i normy

Na potrzeby tego opracowania przyjmujemy następujące definicje:

Standardem jest ujednoczony zbiór cech wzorca danej grupy produktów, do którego może być zbliżony dany produkt.

Normą jest dokument określający cechy wzorca dla danego produktu, który, aby być z nią zgodny, musi je całkowicie spełnić.

Standaryzacja jest ujednoczeniem zbioru cechy danej grupy produktów w celu stworzenia ich wzorca.

Normalizacja jest opracowywaniem dokumentu opisującego cechy wzorca danego produktu.

Inaczej mówiąc, każdy produkt zgodny z normą jest standardowy, ale nie odwrotnie. Jednocześnie w pewnych warunkach i dla określonych organizacji (najczęściej rządowych) zgodność wykorzystywanego przez nie produktu z istniejącą dla niego normą jest obligatoryjna. Zgodność produktu z przyjętym dla niego standardem nie jest wymagana, ale z reguły leży ona w interesie producenta.

W Polsce funkcjonują Polskie Normy (PN) ustanawiane przez Polski Komitet Normalizacji i Miar (PKNiM) oraz normy branżowe i zakładowe, którymi nie będziemy się tutaj zajmować. Normy międzynarodowe są ustanawiane przez międzynarodową organizację normalizacyjną ISO, do której należy też Polska. Normy krajowe członków tej organizacji powinny być zgodne z normami międzynarodowymi.

Dla sprzętu komputerowego i oprogramowania zawartość i zakres norm muszą być często aktualizowane wraz z rozwojem nowych technologii i nowych cech funkcjonalnych tych produktów. Z reguły normy te są zmieniane na wniosek jednej dużej firmy lub stowarzyszenia firm po osiągnięciu sukcesu rynkowego i pojawieniu się różnych mutacji funkcjonalnie podobnych produktów. Procedura opracowywania nowej normy jest z reguły bardzo długa i wymaga pogodzenia wielu rozbieżnych postulatów i ustaleń. Prostsza procedurą jest przyjęcie istniejącej normy międzynarodowej i ustawienie jej jako normy krajowej.

Normy w informatyce

Dla rozwoju informatyki podstawowe są grupy norm (część z nich może być ogłoszonym standardem) dotyczące:

- bezpieczeństwa elektrycznego i promieniowania (monitorów) sprzętu komputerowego;
- rozmiarów i rozmieszczenia połączeń układów i elementów elektronicznych oraz modułów sprzętu;
- sprzętowej realizacji wprowadzania, przesyłania i prezentowania znaków, łącznie ze znakami liter narodowych;
- programowej reprezentacji i przetwarzania (sortowania) kodów znaków, łącznie z kodami znaków liter narodowych;

- składni i znaczenia (semantyki) języków programowania, poleceń systemów operacyjnych, protokołów komunikacyjnych i informatorów struktur wewnętrznych baz danych;
 - krojów kompletów czcionek, symboli graficznych CAD, interfejsów graficznych itp.;
 - formatów standardowych elementów przetwarzanych informacji (np. daty, czasu, oznaczeń dni i miesięcy, symbolu waluty, liczb dziesiętnych itp.);
 - metod atestacji oprogramowania pod względem niezawodności, jakości i zgodności jego funkcjonowania ze specyfikacją oraz z przepisami (np. systemu prowadzenia ksiąg podatkowych zgodnie z aktualnie obowiązującymi przepisami podatkowymi);
 - zasad eksploatacji systemów, czyli m.in. ochrony danych, wymazywania i archiwizowania.
- Niestety, większość wymienionych wyżej norm nie znajduje się w rejestrze Polskich Norm lub też są one przestarzałe w stosunku do stosowanych technologii i potrzeb użytkowników. Z punktu widzenia dzisiejszej sytuacji gospodarczej istnienie niektórych norm jest wręcz szkodliwe.

Nawet w obecnej sytuacji polskiego przemysłu komputerowego brak lub nieaktualność polskich norm dotyczących sprzętu ma istotne znaczenie. Istnieje już możliwość tworzenia organizacji montowni, a w przyszłości produkcji części sprzętu komputerowego. Brak norm powiązanych z normami zachodnimi może stanowić poważny problem i nawet opóźnić powstawanie takich zakładów (a jaki to może być problem, proszę zauważyć różnice we wtyczkach w różnych krajach).

Normy narodowe

Dający się we znaki problem braku polskich norm szczególnie dotyczy wyboru 8-bitowego kodu polskich liter. Istnieją co najmniej trzy przyczyny braku takiej normy. Pierwsza z nich wynika ze ścisłej współpracy polskiego przemysłu komputerowego z przemysłem radzieckim w poprzednich latach. Ze względu na wymagania partnera (żądanie cyrylicy) oraz własną wygodę i oszczędności (jeden rodzaj urządzeń) nie zadbano np. o opracowanie norm na polskie wersje urządzeń zewnętrznych (klawiatur, monitorów, drukarek) oraz sposób reprezentacji polskich liter w kodzie 8-bitowym. Jednocześnie PKNiM był (i nadal jest) instytucją mało efektywną w kreowaniu nowych norm i ich odpowiednim egzekwowaniu. Na przykład od 1 stycznia 1991 obowiązuje według polskiej normy format datowania rok-miesiąc-dzień, którego rzadko się przestrzega.

Trzecia przyczyna stanu rzeczy jest zwana „syndromem żelaznej kurtyny”. Podział świata na Wschód i Zachód znalazł odbicie w międzynarodowym standardzie ISO, dotyczącym 8-bitowych kodów znaków. Przyjęto tam (za zgodą Polski i innych krajów) podział dodatkowych znaków narodowych na część zachodnią i wschodnią Europy (w obu częściach występują znaki alfabetu niemieckiego). Takie rozwiązanie utrudnia np. użytkowanie procesorów tekstów w pisaniu po polsku i po francusku.

Otwarcie w poprzednich latach rynku dalekowschodniego na import do Polski tysięcy mikrokomputerów typu IBM PC spowodowało potrzebę wbudowania w nie znaków polskich liter. Z braku normy, powstało wtedy około 10 różnych

rozwiązania umotywowanych według różnych kryteriów. Tutaj wystarczy stwierdzić, że obecnie przeważająca liczba drukarek oraz sterowników kart Hercules jest przerobiona na tzw. kod mazowiecki. Niestety firmy IBM i Microsoft proponują własny standard dla wszystkich krajów Europy centralno-wschodniej, tzw. CP-852, czyli IBM Latin2, również utrwalając podział Europy na dwie części. Podobne rozwiązanie podziału można znaleźć w opracowywanym 16-bitowym kodzie UNICODE (jeszcze nie mającym oficjalnej normy).

Brak takiej normy ujawnił się też w chwili rozpoczęcia realizacji kontraktu z firmą Bull na dostawę sprzętu minikomputerowego dla systemu podatkowego POLTAX. Po dyskusjach przyjęto za obowiązujący standard ISO 8859/2 (Latin alphabet No. 2), którego Polska jest sygnatariuszem. Była to jedyna możliwość wymuszenia istnienia polskich liter na podstawie uznanego (również przez Francję) międzynarodowego standardu. Jednocześnie PKNiM podpisał 21 marca 1991 Polską Normę (PN-91/T-42115) na kod 8-bitowy, równoważną normie ISO 8859/2. Opisana sytuacja będzie w przyszłości przyczyną wielu nieporozumień i dyskusji.

Następnym problemem okazało się zaprojektowanie polskiej wersji klawiatury. Istniejące normy na układ klawiszy na maszynie do pisania okazały się nieprzydatne ze względu na niepełny zestaw znaków, zmuszający do rezygnacji z wielu istotnych w informatyce znaków graficznych (lewa kreska ukośna, nawiasy kwadratowe i ozdobne itp.). Pojawiły się dwie koncepcje: klawiatura programisty z wprowadzeniem polskich liter z wykorzystaniem klawiszy Alt i Shift oraz klawiatura maszynistki z zamianą istniejących znaków na dodatkowe litery polskie lub ich budowaniem z wykorzystaniem klawisza martwego. Pierwsza koncepcja może być zrealizowana programowo, druga zaś wymaga zmian w sprzęcie. Wobec braku polskiej normy Ministerstwo Finansów zastosowało rozwiązanie pierwsze, z wykorzystaniem dodatkowego klawisza Ext (prawego/zielonego Alt).

Oprócz norm całkowicie związanych z informatyką istnieje wiele norm używanych w innych dziedzinach (urzędy, wydawnictwa, poligrafia itp.), które dla informatyki są normami uzupełniającymi. Przykładem są normy określające formaty standardowych elementów (np. zapis liczb) występujących w wielu produktach programowych. Brak jest norm określających kolejność sortowania (czy najpierw litery wielkie, czy małe?) oraz skrótów nazw dni tygodni i miesięcy. Brakuje też ustalenia formatu adresów, nazw ulic (szczególnie przy zapisie nazw typu: ul. Bitwy Warszawskiej 1920, aleje Marszałka Józefa Piłsudskiego – powodujących konieczność przeszukiwania słowników synonimów), pełnych numerów telefonów (np. + 48.22.123456), faksów i teleksów, poczty elektronicznej itp. Braki te, przy zdecentralizowanym i często dublowanym tworzeniu różnego typu systemów informatycznych, uniemożliwią w przyszłości prostą wymianę informacji oraz działanie w sieci informatycznej.

Klasyfikacje

Standardami są też klasyfikacje rzeczy, instytucji, dziedzin itp. Dotychczas w Polsce korzystano z klasyfikacji własnych, w dużym stopniu uzgadnianych w ramach RWPG. Obecnie jednak rozpoczyna się przechodzenie na odpowiednie klasyfikacje EWG. Na przykład stosuje się już klasyfikację EWG w zakresie jednostek chorobowych, Ministerstwo Współpracy z Zagranicą stosuje klasyfikację wyrobów EWG zamiast SWW. GUS zmienia charakterystykę podmiotów gospodarczych w systemie REGON na układ EWG. Planuje też do roku 1993 zastąpienie SWW odpowiednią klasyfikacją EWG.

Zmiany te są bardzo kosztowne. Nie wystarczy bowiem wymienić zawartości odpowiednich pól w bazach danych. Konieczne będzie nawet nowe uporządkowanie magazynów materiałów według nowych podziałów, które w klasyfikacji

EWG są w innym niż dotychczas stosowanym układzie cech. Dopasowanie się do Europy będzie tutaj sporo kosztować.

Normy atestacji i eksploatacji systemów

W rozdziale 4 pojawia się zalecenie wprowadzenia odpowiedzialności producenta sprzętu i oprogramowania za jakość i zgodność ze specyfikacją funkcjonowania danego produktu. Przyjęcie tego zalecenia wymaga wprowadzenia norm na metody atestacji sprzętu i oprogramowania oraz wskazania, kto i na jakich warunkach może wykonywać takie atestacje i wydawać odpowiednie zaświadczenia. Jednocześnie staje się konieczne opracowanie odpowiednich aktów prawnych określających dziedziny, w których wykorzystywane oprogramowanie musi mieć odpowiedni atest.

Nie istnieją też w Polsce normy eksploatacji systemów (a raczej systemików) informatycznych działających na sprzęcie mikrokomputerowym i w sieciach lokalnych. Systemy funkcjonujące w ośrodkach informatycznych na „dużych” maszynach wymierają. Jednocześnie trzeba sobie zdawać sprawę z niebezpieczeństw wynikających ze zgromadzenia dużych zbiorów informacji coraz powszechniej dostępnych. Szczególna uwaga powinna zostać zwrócona na problemy ochrony danych przed nieuprawnionym użytkownikiem, możliwości zniszczenia danych (wprowadzenie wirusów) (por. rozdz. 11) oraz na wynikające stąd zasady archiwizowania danych i wymazywania danych utajnionych.

Wszystkie te zasady muszą być niestety opracowane na nowo, ze względu na ogromny postęp technologiczny. Oczywiście możemy (i należy to czynić) skorzystać z zasad stosowanych na zachodzie i w USA, z tym jednak, że tego typu informacji nie będzie łatwo uzyskać.

Spolszczanie oprogramowania

Znajdujemy się tuż przed powszechnym spolszczaniem zachodniego (czytaj: amerykańskiego) oprogramowania. Nacjonalizacja oprogramowania odbywa się na czterech poziomach:

- I. Opracowanie firmowej dokumentacji w danym języku, co wymaga istnienia ustalonej terminologii. Przy braku polskiej literatury fachowej dyskusje nad terminologią stają się trudniejsze.
- II. Udostępnienie możliwości wprowadzania, przetwarzania i prezentowania polskich znaków oraz daty, czasu, skrótów dni, miesięcy, liczb dziesiętnych itp. zgodnie z określonymi krajowymi normami (jeżeli istnieją).
- III. Przetłumaczenie wprowadzanych przez oprogramowanie komunikatów, ostrzeżeń oraz innych wiadomości. Zakładamy, że problemy terminologiczne zostały już pokonane.
- IV. Przetłumaczenie całego interfejsu użytkownika oraz udostępnienie możliwości wprowadzania poleceń w języku narodowym.

Dopiero po realizacji prac na wszystkich czterech poziomach możemy mówić o pełnej wersji narodowej. Na razie jeszcze nie znamy przykładu spełniającego podane wymagania. Co więcej, w wielu firmach wykonano lub kończy się już wersje czeskie, węgierskie oraz rosyjskie, pozostawiając wersję polską na koniec. Przyczyna takiego stanu rzeczy leży w komercyjnym podejściu firmy szukającej inwestora opłacającego koszty nacjonalizacji oprogramowania. Druga przyczyna wynika z braku praw ochrony oprogramowania (por. rozdz. 12).

Standard sprzętu i oprogramowania

Obserwując kłopoty agencji EWG z ujednolicaniem funkcjonowania systemów informatycznych przed zjednoczeniem Eu-

ropy, warto już teraz wykorzystać ich doświadczenia, ustalając uogólniony standard sprzętu i oprogramowania. Standard ten powinien być brany pod uwagę przy wszelkich modyfikacjach istniejących systemów, a także przede wszystkim przy projektowaniu nowego systemu. Obecnie zarówno rząd USA, jak i agencje EWG preferują standard zwany „systemem otwartym”, o następujących cechach:

- dowolny sprzęt komputerowy, umożliwiający funkcjonowanie systemu operacyjnego typu UNIX;
- system operacyjny typu UNIX zgodny ze standardem X/POSIX;
- oprogramowanie systemu zarządzania relacyjnymi bazami danych wykorzystujące język zapytań zgodny ze standardem SQL;
- oprogramowanie komunikacyjne oparte na protokole X.25 oraz standardzie OSI/ISO. (por. rozdz. 7).

Zagrożenia

Przy lawinowej wymianie kilkunastu dużych centrów obliczeniowych na tysiące mikrosystemów istnienie pewnych zasad standaryzacji jest podstawą jakiegokolwiek strategii rozwoju informatyki. Jedynie bowiem wprowadzenie dobrze przemyślanych standardów lub nawet norm oraz ich egzekwowanie pozwala w drodze częściowych zakupów w przyszłości powiązać informacyjnie te systemy przy minimalizacji wydatków na różnego typu konwertery sprzętu i programowe. Należy też zdać sobie sprawę z trudności eksploatacji tak różnorodnego sprzętu i oprogramowania. Jednocześnie brak norm opóźnia powstawanie wersji polskich oraz, co można powiedzieć z całą odpowiedzialnością, lekceważenie naszego rynku przez firmy zachodnie.

Istnieje już groźna sytuacja z 8-bitowym kodem polskich liter. Wprowadzenie polskiej wersji normy ISO było mocno spóźnione. Brakowało jednoznacznego stanowiska w sprawie kodów dla mikrokomputerów. Dlatego też Microsoft i IBM samodzielnie wprowadziły swój kod, które zmusi nas do powtórnego przeprogramowania tysięcy kostek pamięci EPROM w drukarkach i sterownikach monitorów (w cenie ok. 0,5 mln zł. za jeden komplet). Zwracamy też uwagę na zagrożenia związane z brakiem atestacji wielu systemów informatycznych oraz brakiem zasad eksploatacji. Używanie niepewnych systemów będzie przynosić ogromne straty finansowe oraz społeczne.

Zalecenia

1. Jest konieczna weryfikacja norm dotyczących informatyki pod względem ich aktualności i kompletności w nowej sytuacji gospodarczej, kierunków współpracy oraz nowych technologii. Normy już niepotrzebne należy wyciąć.

2. Jest niezbędne opracowanie – we współpracy z jednostkami naukowymi i firmami komputerowymi (w tym prywatnymi!) – nowych standardów i norm dotyczących sprzętu komputerowego, 8-bitowych kodów polskich znaków, układu klawiatur, protokołów komunikacyjnych i innych norm uzupełniających.

3. Należy podjąć szerszą współpracę z międzynarodowymi (ISO) i narodowymi (np. ANSI) organizacjami standaryzacyjnymi w celu prezentowania i uwzględnienia w przyszłych normach (np. UNICODE) naszych poglądów i wymagań. Szczególnie celowe jest podjęcie współpracy w Grupie Heksagonalnej w kierunku zunifikowania wymagań dotyczących kodów znaków narodowych, co pozwoli na łatwiejsze negocjacje z firmami zachodnimi oraz z ISO.

4. Należy zainicjować prace (np. w ramach projektów rządowych finansowanych przez KBN) na temat metod atestacji jakości oprogramowania oraz zasad eksploatacji systemów

informatycznych z uwzględnieniem ochrony danych, zasad aktualizacji i archiwizowania. Na tej podstawie będzie możliwe opracowanie odpowiednich norm i powołanie instytucji nadzoru nad jakością systemów informatycznych wdrażanych w administracji państwowej oraz w innych jednostkach o dużym stopniu ryzyka działania (np. obronność, banki, komunikacja, medycyna itp.).

5. Podobne prace należy zainicjować w celu zbadania zakresu koniecznych działań i ich kosztów związanych z zastosowaniem w gospodarce klasyfikacji EWG zastępujących klasyfikacje dotychczas używane. Równocześnie jest konieczne przetłumaczenie, opracowanie i rozpropagowanie tych klasyfikacji w celu umożliwienia przygotowania i dokonania zmian w istniejących i projektowanych systemach informatycznych.

6. Zawierając kontrakty na informatyzację administracji i innych jednostek państwowych, należy dołączyć wymagania na dostarczenie polskich wersji oprogramowania w takiej postaci, żeby końcowy użytkownik mógł wykorzystywać cały system tylko w języku polskim.

7. Należy szeroko wyjaśnić i propagować koncepcję systemu otwartego, tak aby przyszłe systemy mogły być łatwo wymienne w poszczególnych warstwach, w pełni dyspozycyjne oraz łatwo przyłączalne.

Rozdział 11

Przestępczość informatyczna

Wprowadzenie

W miarę postępu zastosowań informatyki rozwijają się towarzyszące zjawiska patologiczne. Najpoważniejsze z nich – przestępstwa informatyczne – wymagają systematycznego przeciwdziałania ze strony państwa. Jak wobec każdej klasy przestępstw, należy więc podjąć odpowiednie działania legislacyjne (określenie czynów przestępczych i ustalenie wymiaru odpowiedzialności za nie) i prewencyjne oraz wdrożyć niezbędne środki ścigania.

Przestępstwa informatyczne można z grubsza podzielić na dwie kategorie:

- przestępstwa popełniane z użyciem środków informatyki;
- przestępstwa wobec systemów informatycznych.

Klasyfikacja ta nie jest rozłączna, gdyż np. wielu wykroczeń wobec systemów informatycznych dokonuje się z użyciem środków informatyki: wprowadzamy ją jednak dlatego, że o ile przestępstwa popełnione z użyciem środków informatyki mają swe naturalne odpowiedniki klasyczne, ujęte w Kodeksie Karnym, o tyle przestępstwa informatyczne (wobec systemów) na ogół nie mają takich odpowiedników, a podciąganie ich pod zbliżone kategorie klasyczne prowadzi z reguły do zaniżonej oceny szkodliwości. Z tego właśnie powodu w wielu krajach wprowadza się specjalne przepisy prawno-karne dla tej kategorii czynów. Dla przykładu, wedle brytyjskiego The Computer Misuse Act, obowiązującego od sierpnia 1990 r., przestępstwem karnym (crime) jest każde wejście (access) do systemu informatycznego dokonane bez zezwolenia i to niezależnie od tego, z jakim zamiarem dokonano owego wejścia, oraz każda modyfikacja programu lub danych zapisanych w pamięci lub na nośniku zewnętrznym, na którą nie miało się jawnego zezwolenia. W świetle tej i podobnych norm prawnych samo wchodzenie do systemów informatycznych dla zabawy i z ciekawości staje się przestępstwem karnym, niezależnie od tego, czy spowodowało jakieś szkody.

Przestępstwa dokonywane z użyciem środków informatyki

Przestępstwa te mogą dotyczyć nie tylko wszystkich dziedzin, w których stosuje się informatykę, lecz także tych, w których nie jest ona jeszcze powszechnie stosowana; bywa niekiedy i tak, że pionierskie zastosowania informatyki w pewnej dziedzinie służą celom przestępczym. Oprócz naturalnych przestępstw i wykroczeń, takich jak defraudacja, wyłudzenia, oszustwa podatkowe, nierzetelna księgowość, fałszerstwa dokumentów i danych itp., użycie środków informatyki pozwala dokonywać przestępstw polegających na nielegalnym pozyskiwaniu, gromadzeniu i kojarzeniu informacji zastrzeżonych (poufnych i tajnych), w szczególności zaś na dedukcyjnym pozyskiwaniu zastrzeżonych wiadomości z cząstkowych wiadomości ogólnodostępnych. Aczkolwiek prawie wszystkie takie przestępstwa znane są w swojej przedinformatycznej postaci, kryminalne użycie środków informatyki powoduje, że znacznie trudniej wykryć je metodami klasycznymi; czasami w ogóle nie można tego dokonać bez zastosowania specjalnych, także informatycznych, metod kontroli i dochodzenia. Wynika stąd konieczność wielostronnych działań, których inicjatorem, a często także realizatorem, powinny być odpowiednie instytucje publiczne.

1. Należy odpowiednio przeszkolić tych pracowników administracji państwowej i municypalnej oraz wymiaru sprawiedliwości, którzy z racji pełnienia swych obowiązków służbowych powinni umieć rozpoznawać objawy przestępstw informatycznych. Szkolenie to, zapewne na zróżnicowanych poziomach szczegółowości, powinno osiągnąć dwa cele; uczulił rzeczoną grupę pracowników na problem przestępczości informatycznej i nauczył ich właściwych metod postępowania wobec zagrożenia takimi przestępstwami. Jest rzeczą kapitalnej wagi, by obawa przed przestępstwami informatycznymi nie zahamowała rozwoju informatycznych form prowadzenia księgowości, dokumentacji handlowej, administracji, bankowości itp., ani też nie spowodowała dwutorowości takich działań, np. księgowania ręcznego dla kontroli i informatycznego dla użytku własnego; oprócz oczywistej nieefektywności, taka dwutorowość jest właśnie najbardziej kryminogenna. Pracownicy administracji państwowej powinni być świadomi zagrożeń, ale nie bać się ich (gdyż lęk zbyt łatwo przekształca się w hamulec postępu).

2. Należy przygotować zestaw przepisów odnośnie do oprogramowania, sprzętu komputerowego, łączności i trybu użytkowania systemów informatycznych, ułatwiających informatyczną kontrolę legalności czynności wykonywanych z użyciem tych systemów i informatyczne wykrywanie działań nielegalnych. Przepisy te, o zróżnicowanym stopniu szczegółowości i kompletności przewidywanej kontroli, powinny obowiązywać w tych wszystkich systemach, które są prowadzone na użytek lub zlecenie organów administracji państwowej i municypalnej, a także wszędzie tam, gdzie organa administracji państwowej na mocy ustawy prawem rutynowej kontroli (np. w powszechnych bankach i towarzystwach ubezpieczeniowych). Stosowanie tych przepisów w pozostałych systemach informatycznych nie musi być wymagane przez prawo; należy jednak przypuszczać, że staną się one normami *de facto*.

3. Należy podjąć budowę informatycznych narzędzi kontroli i inspekcji systemów informatycznych oraz dostarczanych przez nie danych. Narzędzia te powinny być oddane do dyspozycji organów administracji, kontroli i inspekcji. Z oczywistych względów zestaw takich narzędzi nie może być stały; przeciwnie, powinien być często uzupełniany, modyfikowany i zmieniany, z zachowaniem jednolitej metodyki użytkowania. Wydaje się więc, że należy powołać specjalną placówkę złożoną z wysokiej klasy informatyków, której zadaniem byłoby systematyczne tworzenie informatycznych narzędzi kontroli i inspekcji systemów informatycznych oraz ich działalności. Wspomniane powyżej przepisy dotyczące ułatwienia kontroli i inspekcji powin-

ny być uzgodnione z tą placówką, a może nawet przez nią opracowane.

4. W policji państwowej, NIK i GUC należy powołać wyspecjalizowane komórki do walki z przestępstwami informatycznymi. Jest smutną prawidłowością rozwoju zastosowań informatyki, że komórki te będą miały coraz więcej pracy.

5. Należy powołać niezależną placówkę atestacji oprogramowania, której zadaniem będzie stwierdzenie, czy poszczególne pakiety użytkowe spełniają przepisy o ułatwieniach kontroli i inspekcji, i czy są rzetelne. Gdyby atest tej placówki miał być obowiązujący przy zakupach oprogramowania dla administracji państwowej i municypalnej, powinna ona być finansowana z budżetu państwa. Należy podkreślić, że sam fakt posługiwania się atestowanym oprogramowaniem nie jest jeszcze dostateczną gwarancją rzetelności i legalności wyników pracy systemu informatycznego i pod żadnym pozorem nie może być traktowany jako zwolnienie z kontroli i inspekcji, gdyż przestępstwo może być popełnione poza bezpośrednim nadzorem atestowanego pakietu (np. przy pomocy odpowiednio zmodyfikowanego systemu operacyjnego albo wskutek działań osób obsługujących system; może też być rezultatem przestępstwa popełnionego względem samego systemu, np. rezultatem bazprawnej modyfikacji atestowanego pakietu). Stosowanie atestowanych pakietów ułatwia jednak kontrolę i inspekcję, i dlatego – zakładając rzetelność samej atestacji – można się spodziewać powszechnego zaakceptowania procedury atestacji.

Przestępstwa wobec systemów informatycznych

Przestępstwa te polegają na nieuprawnionej ingerencji w działanie systemów, nieuprawnionym korzystaniu z ich zasobów (obliczeniowych, pamięciowych i transmisyjnych), zniekształcaniu treści informacji zapisanych w pamięci komputerowej i na nośnikach zewnętrznych, nieuprawnionym kopiowaniu takiej informacji i/lub wykorzystywaniu jej bez zezwolenia, a także na niszczeniu systemów, ich elementów, powiązań (fizycznych lub logicznych) i struktur. Do przestępstw wobec systemów informatycznych należą więc przykładowo: rozpowszechnianie tzw. wirusów komputerowych, bezprawne korzystanie z usług systemu, nieuprawnione odczytywanie danych, zmiana parametrów systemowych, samowolne przyłączenie drukarki, podsłuch transmisji, czytanie cudzej poczty elektronicznej, nieuprawnione przyłączanie swego komputera do sieci. Ponieważ wiele z czynności mogących spowodować nieobliczalną szkodę dla legalnych dysponentów i uprawnionych użytkowników systemu informatycznego może być prezentowane jako niewinna zabawa lub dociekania badawcze, a w każdym razie jako czynności wykonywane bez przestępczych intencji, jest rzeczą pierwszorzędnej wagi ustawowe uznanie ich za przestępstwo niezależnie od skutków i intencji.

Przeciwdziałanie przestępstwom wobec systemów informatycznych – podobnie jak zapobieganie i zwalczanie przestępstw informatycznych – wymaga wielu przedsięwzięć. Nie wszystkie z tutaj wymienionych mają zastosowanie do wszystkich systemów, bezpieczeństwo nie wszystkich bowiem systemów jest tak samo ważne. Podając w miarę kompletną listę staramy się przedstawić całość zagadnienia, w którym rządowi przypada rola kluczowa z dwu względów:

- wiele państwowych systemów informatycznych jest szczególnie narażonych na działania przestępcze;
- na rządzie spoczywa obowiązek zainicjowania i wdrożenia dobrej praktyki ochrony przed działaniami przestępczymi wobec systemów informatycznych.

1. Należy doprowadzić do ustawienia normy prawnej ustalającej, że każdy akt nieupoważnionego (tj. bez wyrażonego przyzwolenia) wejścia do systemu informatycznego i każda nieupoważniona modyfikacja programu lub danych komputerowych (w pamięci lub na nośniku zewnętrznym) jest czynem

karalnym bez względu na to, jakie były intencje i skutki tego czynu, oraz bez względu na to, czy nastąpiło przy tym naruszenie praw autorskich, czy nie. Przez wejście do systemu informatycznego należy przy tym rozumieć nie tylko aktywną ingerencję w stan pamięci i rejestrów komputerów, urządzeń peryferyjnych i kanałów łączności, lecz także odczytywanie lub kopiowanie stanu pamięci, rejestrów, wskaźników i sygnałów w komputerach, urządzeniach peryferyjnych i kanałach łączności, a także wprowadzanie jakichkolwiek informacji do pamięci, rejestrów i kanałów łączności systemu informatycznego. Za nieuprawnione wejście przyjmuje się także przyzwolenie. Czynem karalnym jest więc np. samo już wprowadzenie wirusa komputerowego do systemu, bez względu na to, czy zostanie on kiedykolwiek uruchomiony i czy spowoduje to jakiegokolwiek szkody (karalne jest więc wprowadzenie żartobliwych wirusów). Tak surowe traktowanie nieupoważnionego wejścia do systemu i wszelkich nieupoważnionych modyfikacji programów i danych ma na celu obezwładnienie obrony typu: „nie chciałem niczego złego zrobić, chciałem tylko zażartować” oraz „włamałem się do systemu po to, żeby udowodnić, iż jest to możliwe”. Zdecydowana ochrona prawna nienaruszalności systemów informatycznych i informacji w nich zawartej jest absolutnie konieczna w celu przecięcia nie kończących się debat na temat co wolno, a czego nie. Ryzyko ogromnych szkód społecznych wynikających z tzw. żartów w odniesieniu do systemów informatycznych jest zbyt wielkie na to, by zezwolić na liberalne traktowanie „agresji bez złej woli”.

2. Należy opracować zbiór wymagań względem bezpieczeństwa systemów, programów i danych informatycznych odpowiadających poszczególnym stopniom hierarchii zastrzeżeń („do użytku wewnętrznego”, „poufne”, „tajne” itp.). Wymagania te powinny precyzować zasady odnośnie do oprogramowania, sprzętu, kanałów łączności i urządzeń peryferyjnych systemów informatycznych zasady szfrowania informacji w systemach informatycznych i ich elementach, zasady prowadzenia usług powierniczych dla dystrybucji kluczy szfrowania, zasady ochrony publicznych sieci informatycznych i odpowiedzialności ich operatora za skutki naruszenia bezpieczeństwa, zasady fizycznej ochrony systemów informatycznych oraz magazynów programów i danych przed nieupoważnionym dostępem. Przy opracowywaniu tych zasad należy posłużyć się normami i zaleceniami ISO, CCITT oraz EWG, tak aby zasady wprowadzone w Polsce nie były słabsze od międzynarodowych.

3. Nawiązując do zasad ochrony (por. powyżej) należy ustawowo określić stopnie zabezpieczenia dla poszczególnych państwowych i rządowych systemów informatycznych, a także dla tych systemów, których bezpieczeństwo ma bezpośredni wpływ na bezpieczeństwo (fizyczne, finansowe i moralne) obywateli (np. systemy bankowe, powszechnych ubezpieczeń, bazy danych personalnych, medycznych itp.). Uzyskanie licencji na prowadzenie działalności gospodarczej powinno wiązać się z podjęciem zobowiązania prowadzenia systemu informatycznego określonej klasy bezpieczeństwa, państwu zaś powinno przysługiwać prawo weryfikacji bezpieczeństwa stosowanych systemów (na wzór inspekcji przeciwpożarowej czy budowlanej).

4. Należy opracować zalecenia dotycząca przeciwwaryjnego bezpieczeństwa systemów i zbiorów danych informatycznych (przerwy w zasilaniu, awarie pamięci pomocniczej, omyłki operatorskie). Zalecenia takie powinny być opracowane łącznie z zasadami zabezpieczeń przed czynami przestępczymi, gdyż dopiero ich suma stanowi o rzeczywistym bezpieczeństwie systemów.

5. Należy powołać placówkę atestacji bezpieczeństwa systemów informatycznych działającą co najmniej w dziedzinie systemów państwowych, a lepiej – świadczącą usługi atestacyjne dla wszystkich chętnych, co pozwoliłoby częściowo zmniejszyć obciążenie budżetu państwa z tytułu jej prowadzenia. Atestacja odpowiedniej klasy bezpieczeństwa systemów państwowych i municypalnych powinna być obowiązkowa, podobnie jak atestacja systemów użytku i sieci teleinformatycznych.

6. Należy powołać placówkę weryfikacji bezpieczeństwa systemów informatycznych i sieci teleinformatycznych. Jej pracownicy powinni prowadzić okresowe (nie zapowiadane) inspekcje bezpieczeństwa systemów podlegających nadzorowi rządu.

7. Należy powołać wyspecjalizowaną komórkę policji państwowej do zwalczania przestępstw wobec systemów informatycznych.

Uwaga. To, czy wymienione komórki i placówki do spraw przestępstw informatycznych i przestępstw wobec systemów informatycznych mogą stanowić jedną całość organizacyjną, powinno być przedmiotem dodatkowej analizy. W każdym razie nie wydaje się celowe łączenie zadań inspekcyjno-kontrolnych i dochodzeniowych w jednej organizacji.

W niniejszym opracowaniu nie wspominamy o przestępstwach dotyczących naruszeń praw własności intelektualnej w odniesieniu do oprogramowania i topologii układów elektronicznych, gdyż nie są to sensu stricto przestępstwa informatyczne, a odpowiednią ochronę prawną stwarza prawo autorskie i prawo patentowe. Należy jednak podkreślić pilną konieczność znalezienia odpowiednich ustaw (w duchu zaleceń EWG), tak aby wyraźnie odpowiadały specyficznym wymaganiom informatycznym. W tej sprawie PTI wielokrotnie zajmowało publiczne stanowisko i przygotowało projekty odpowiednich zapisów prawnych (por. rozdz. 12). Należy dodać, że efektywne zwalczanie piractwa oprogramowania wymaga powołania wyspecjalizowanego aparatu ścigania – na wzór tego, który podjął walkę z piratami nagrań fonicznych i wizyjnych.

Rozdział 12

Proponowane regulacje prawne

W niniejszym opracowaniu, stanowiącym propozycję strategii rozwoju informatyki i jej zastosowań w Polsce, zwracano uwagę, że realizacja tej strategii wymaga zarówno działań władczych ze strony rządu, jak i inicjatywy ustawodawczej w celu ukształtowania stanu prawnego stwarzającego warunki właściwego rozwoju oraz wykorzystania systemów informatycznych.

Treścią niniejszego, zwięzłego rozdziału jest usystematyzowane wyliczenie propozycji dotyczących zagadnień wymagających, zdaniem autorów opracowania, regulacji prawnej.

1. Główny obszar proponowanej regulacji ustawowej dotyczy systemów informatycznych o znaczeniu publicznym, zasad ich tworzenia, dopuszczania do eksploatacji oraz wykorzystania gromadzonej i przetwarzanej informacji. Przez systemy informatyczne o znaczeniu publicznym należy rozumieć systemy informatyczne:

– tworzone przez organy administracji rządowej i samorządowej w celu wspomaganie wykonywania przez te organy ich funkcji władczych;

– tworzone przez inne podmioty, o ile charakter, zakres i ilość gromadzonych informacji, a także sposób przetwarzania, powodują, że wykorzystywanie systemu może mieć wpływ na sytuację prawną osób trzecich.

Regulacja taka powinna obejmować zagadnienia:

– ograniczeń w odniesieniu do zakresu i sposobu agregowania informacji;

– praw i zakresu obowiązków osób, których dotyczy informacja przetwarzana w systemie;

– obowiązków z zakresu uprawnień osób, które administrują systemem;

- zasad ochrony danych oraz ich udostępniania;
- dopuszczalności wykorzystania informacji uzyskanych w wyniku działania systemu (utrwalonych w postaci wydruku) jako dokumentu;
- wymagań w odniesieniu do technologii tworzenia systemu, w tym wymagań normalizacyjnych i homologacyjnych dotyczących modułów sprzętu i oprogramowania, wykorzystywanych w tworzeniu systemu;
- warunków technicznych i organizacyjnych dopuszczenia systemu do eksploatacji, w tym zasad atestacji systemu oraz podmiotów uprawnionych do jej przeprowadzania;
- odpowiedzialności karnej osób zarówno wykorzystujących system informatyczny w celu popełnienia przestępstwa, jak i ingerujących bezprawnie w działanie systemu.

Pierwsze cztery z wymienionych powyżej zagadnień tworzą nieuregulowaną dotychczas w Polsce sferę **zasad gromadzenia informacji, ochrony danych oraz ochrony dóbr osobistych potencjalnie zagrożonych w wyniku eksploataowania systemów informatycznych**. Następne trzy z wymienionych powyżej zagadnień mieszczą się w sferze regulacji prawnych dotyczących **normalizacji i jakości. Penalizacja określonych czynów mających związek z wykorzystaniem systemów informatycznych** stanowi uznanie faktu, że specyfika systemu informatycznego stwarza nową jakość w społecznych i ekonomicznych skutkach czynów takich i wymaga specjalnych sankcji.

Zarówno dokonanie całościowej regulacji powyższych zagadnień, jak i uregulowanie ich w drodze odpowiednich nowelizacji wielu przepisów, wymaga odrębnego przedsięwzięcia legislacyjnego, które powinno być przygotowane przez specjalny zespół ekspertów.

2. Jednoznacznego rozstrzygnięcia legislacyjnego wymaga kwestia **ochrony prawnej programów komputerowych**. Sprawa ta i jej znaczenie podnoszone były przez PTI w stanowisku z dnia 15 grudnia 1988 r. (w załączeniu tekst stanowiska) oraz przedkładanych organom państwowym opiniach dotyczących kolejnych projektów nowej ustawy o prawie autorskim.

Przedsięwzięcie legislacyjne w tym zakresie powinno uwzględnić opinię ekspertów-informatyków i być wynikiem określonej polityki w stosunkach gospodarczych z zagranicą (por. rozdz. 2).

3. Realizacja przyjętej przez rząd strategii rozwoju informatyki i jej zastosowań wymaga ponadto wydania podporządkowanych tej strategii rozporządzeń wykonawczych.

Rozporządzenia te powinny w szczególności ustalać:

- wynikające z przyjętej strategii **zasady oprogramowania** komputerowego, w odniesieniu do którego brak dotychczas jednoznacznych reguł;
- wynikające z przyjętej strategii **zasady opodatkowania** usług informatycznych, w szczególności opodatkowania wytwarzania oprogramowania oraz obrotu oprogramowaniem gotowym, w odniesieniu do którego brak dotychczas jednoznacznych zasad – w związku z brakiem regulacji statusu oprogramowania jako dobra niematerialnego.

Ochrona prawna programów komputerowych

Stanowisko Polskiego Towarzystwa Informatycznego z 15 grudnia 1988

1. W środowisku Polskiego Towarzystwa Informatycznego jest od dłuższego czasu podnoszona teza o konieczności ustanowienia jednoznacznej ochrony prawnej programów komputerowych. Przemawiają za tym – zgodnie z opinią środowiska – następujące argumenty:

- ustanowienie takiej ochrony jest jednym z koniecznych warunków dla rozwoju krajowego przemysłu oprogramowania, bez którego istnienia nie może być mowy o postępie zastosowań informatyki w Polsce;

- obserwowane w skali masowej zjawisko piractwa w dziedzinie oprogramowania powoduje wiele szkód w gospodarce narodowej na skutek częstego używania oprogramowania niekompletnego i niewłaściwie udokumentowanego lub uszkodzonego w wyniku łamania zabezpieczeń;
- kopiowanie na szeroką skalę oprogramowania zagranicznego bez zgody i woli producentów stanowi łamanie zasad przyjętych w obrocie w ich krajach i jest poważną barierą dla rozwoju poprawnych stosunków handlowych z tymi producentami;
- sytuacja, w której można za darmo otrzymać kopie programów, stwarza społeczne przekonanie, że oprogramowanie nie kosztuje lub kosztuje niewiele. To przekonanie jest błędne i może mieć poważne skutki dla podejmowanych decyzji gospodarczych.

2. Konieczność ustanowienia ochrony prawnej oprogramowania jest ściśle związana z prowadzonymi analizami oraz wypowiedziami przedstawicieli nauki prawa, z których wynika, że obowiązujący w Polsce prawny system ochrony własności intelektualnej (prawo autorskie, prawo wynalazcze) jest w niedostatecznym stopniu przystosowany do tego, aby mógł być podstawą jednoznacznej kwalifikacji programów komputerowych jako przedmiotu ochrony. Fakt ten dostrzeżono w krajach wytwarzających oprogramowanie komputerowe na skalę przemysłową, dokonując stosowanej nowelizacji obowiązującego prawa autorskiego. Sugestie w tym zakresie wynikają ponadto z prac Światowej Organizacji Własności Intelektualnej, której członkiem jest Polska.

3. Nie przesądzając ostatecznych rozwiązań legislacyjnych Polskie Towarzystwo Informatyczne podjęło prace nad zdefiniowaniem podstawowych elementów podmiotowego prawa bezwzględnego (prawo własności programu), mającego za przedmiot programy komputerowe. Podstawowe elementy definicji proponowanej przez Towarzystwo są załączone do tekstu niniejszego stanowiska.

4. Polskie Towarzystwo Informatyczne prezentuje pogląd, że ochrona prawna programów komputerowych, będąc warunkiem koniecznym, nie jest jedynym wystarczającym warunkiem rozwoju przemysłu oprogramowania. Warunkiem tego rozwoju są ponadto: właściwa polityka kredytowa i podatkowa, umożliwiająca odpowiednie inwestycje oraz właściwe dla przedmiotu produkcji zasady rozliczenia jej kosztów. Wprowadzane w kraju nowe zasady prowadzenia działalności gospodarczej oraz nowe zasady polityki finansowej powinny uwzględniać specyfikę produkcji oprogramowania, jej technologię oraz charakter powstających produktów. Możliwość uwzględnienia tej specyfiki jest ściśle związana z uregulowaniami dotyczącymi stosunków własnościowych, tj. prawa własności programów.

5. Wysokość opłat licencyjnych na rzecz producentów z drugiego obszaru płatniczego, wymaganych od chwili wprowadzenia ochrony prawnej oprogramowania, przytaczana jako argument przeciwko jej ustanowieniu, jest – zdaniem Towarzystwa – w znacznym stopniu wyolbrzymiona. Jest ona oceniana przy założeniu konieczności wykorzystania wszystkich producentów programowych (oraz rozmaitych ich wersji), które pojawiły się na rynku krajowym w wyniku pirackiego kopiowania. Potrzeby zastosowań informatyki wymagają znacznie mniejszej liczby tych produktów, sprowadzanych jedynie w określonych wersjach. Przy takim założeniu wysokość tych opłat należy szacować na znacznie niższym poziomie. Konieczność wnoszenia opłat może stanowić natomiast istotny czynnik dla prowadzenia ekonomicznie uzasadnionej selekcji oprogramowania importowanego i doprowadzić do praktyki dokonywania transakcji z mniejszą liczbą producentów, co może mieć korzystny wpływ w zakresie typizacji oprogramowania produkowanego w kraju.

1. Przedmiotem prawa własności programu jest każdy program komputerowy ustalony w postaci:
 - a) kodu programu w wersji do bezpośredniego wykonania przez komputer,
 - b) tekstu programu w języku symbolicznym lub innej postaci umożliwiającej automatyczne wygenerowanie kodu programu w wersji do bezpośredniego wykonania przez komputer.
- 1.1. Przedmiotem prawa własności programu jest również każdy szczegółowy opis, który umożliwia proste odtworzenie którejkolwiek z postaci wymienionych w p. 1.
2. Podmiotem prawa własności programu jest osoba, która poniosła nakłady na opracowanie programu. W szczególności, jeżeli program został opracowany w wykonaniu umowy o pracę lub umowy o dzieło, prawo własności programu służy pracodawcy lub zamawiającemu.
- 2.1. Jeżeli kilka osób poniosło nakłady związane z opracowaniem programu, dla określenia prawa każdej z osób stosuje się odpowiednio przepisy prawa cywilnego dotyczące współwłasności.
- 2.2. Domniemywa się, że prawo własności programu służy osobie, której nazwa jest uwidoczniiona w szczegółowym opisie programu lub która zostaje uwidoczniiona w zewnętrznych przejawach działania programu (np. na ekranie, wydruku lub rysunku).
- 2.3. Bezpośrednim twórcą programu służy prawo ochrony dóbr osobistych, zgodnie z zasadami prawa cywilnego. Nie obejmuje ono jednak prawa do decydowania o sposobie wykorzystania programu przez właściciela prawa do tego programu.
3. Prawo własności programu obejmuje prawo do:
 - a) wyłącznego rozporządzania programem,
 - b) pobierania przychodów z tytułu eksploatacji programu.
- 3.1. Prawo własności programu może być w całości lub części zbyte innej osobie.
4. Poszczególne uprawnienia w zakresie prawa własności programu mogą być udzielone przez właściciela prawa do programu innej osobie w drodze umowy licencyjnej.
5. Korzystanie z cudzego programu może odbywać się wyłącznie na podstawie umowy zawartej z właścicielem prawa do tego programu lub osobą uprawnioną wg zasad określonych w p. 4.
6. Naruszeniami prawa własności programu są ponadto:
 - a) kopiowanie programu,
 - b) odtwarzanie tekstu programu w języku symbolicznym lub opisu programu na podstawie kodu programu w wersji do bezpośredniego wykonania,
 - c) wykorzystywanie istotnych rozwiązań zawartych w programie w opracowaniu innego programu, bez pisemnej zgody właściciela prawa do tego programu lub osoby uprawnionej wg zasad określonych w p. 4
7. Opublikowanie przez właściciela prawa do programu szczegółowego opisu tego programu w książce lub czasopiśmie jest uważane za wyrażenie zgody na swobodne wykorzystywanie programu oraz zawartych w nim rozwiązań.
8. Usuwanie nazw osoby lub osób, którym przysługuje prawo własności programu, oraz stwarzanie w jakikolwiek sposób przekonania, że prawo to przysługuje innej osobie lub osobom, są ścigane karnie.
9. Właściciel prawa do programu lub osoba upoważniona wg zasad określonych w p. 4 może żądać od każdego, kto narusza zasady określone w p. 5 i 6, zaniechania tych naruszeń oraz wydania korzyści materialnych.

Kluczowa rola zastosowań informatyki oraz przemysłu i usług informatycznych w nowoczesnym państwie wymaga dobrze określonej strategii w tych dziedzinach. Rząd – ze względu na swe uprawnienia konstytucyjne – jest właściwym organem państwowym, który taką strategię powinien zaplanować i wprowadzić w życie. Brak działań rządowych spowoduje, że nieunikniony rozwój zastosowań informatyki w sferze publicznej będzie odbywał się w sposób chaotyczny i nie kontrolowany, a Polska stanie się areną ścierania się interesów innych państw, realizujących własne, często wzajemnie sprzeczne strategie. W rezultacie współdziałanie różnych publicznych systemów informatycznych w Polsce będzie znacznie utrudnione, a inwestycje informatyczne okażą się nieefektywne. Zmniejszą również szanse uzyskania przez Polskę odpowiedniego miejsca w międzynarodowym podziale pracy w dziedzinie informatyki.

Proponowana strategia polega na przyjęciu zasady priorytetu zastosowań informatyki nad produkcją środków informatycznych ze wszystkimi konsekwencjami tego podejścia, łącznie z tą, że producenci sprzętu informatycznego i oprogramowania są finansowani wyłącznie przez konsumentów zastosowań informatyki. Taka strategia jest odwróceniem zasad obowiązujących w Polsce w ostatnim dwudziestolecu.

Funkcja rządu w określeniu strategii rozwoju informatyki ma dwa komplementarne aspekty:

– decyzyjny, związany z podejmowaniem konkretnych, jednostkowych decyzji strategicznych w dziedzinach: polityki obronnej, ochrony bezpieczeństwa państwa i obywateli, polityki zagranicznej, edukacji, stymulowania badań i polityki kadrowej, tworzenia infrastruktury informacyjnej, informatyzacji administracji państwowej, budowy publicznej sieci komputerowych oraz przeciwdziałania przestępczości informatycznej;

– prawodawczy, związany z ustaleniem ogólnie obowiązujących zasad w odniesieniu do polityki gospodarczej, standaryzacji, atestacji systemów informatycznych, prawnej ochrony oprogramowania oraz ochrony dóbr osobistych obywateli w związku z rozwojem systemów informacyjnych.

W poszczególnych rozdziałach *Propozycji strategii rozwoju informatyki i jej zastosowań w Rzeczypospolitej Polskiej* zawarto szczegółowe omówienie niezbędnych działań w niemal wszystkich wymienionych dziedzinach, wraz z odpowiednią argumentacją. Wyjątkami są: polityka obronna oraz ochrona bezpieczeństwa państwa i obywateli; dziedziny te wymagają odrębnych opracowań, aczkolwiek ogólne działania proponowane w niniejszym opracowaniu mogą być również w nich stosowane, jeśli nie przeszkadza temu konieczność zachowania tajemnicy państwowej.

Podstawowe zadania strategiczne na arenie międzynarodowej wiążą się z rozpoczynającym się procesem strukturalnej integracji Polski z Europą zachodnią. Jednym z pożądanych skutków takiej integracji powinien być wybór Polski jako regionalnego centrum infrastruktury informatycznej, stanowiącego pomost między Europą zachodnią a ZSRR. W tym celu rząd polski powinien być gotowy do wniesienia poważnego wkładu w budowę takiego centrum. Wkładem tym mogą być znaczne zasoby kadrowe informatyków znających specyfikę rynku wschodnioeuropejskiego i nieźle zaznajomionych z realiami informatyki zachodniej, istniejące możliwości szkolenia kadr oraz pozostały jeszcze przemysł informatyczny, który – zasilony technologiami i kapitałem zagranicznym – może dostosować się do nowych zadań. Ponadto należy podjąć działania gospodarcze i polityczne zachęcające duże koncerny informatyczne do lokowania swych central wschodnioeuropejskich i ośrodków badawczych w Polsce, a nie w Austrii lub na Węgrzech, jak dzieje się obecnie.

Integracja z Europą zachodnią wymaga także stałego uczestnictwa polskich delegatów w pracach organizacji międzynarodowych zajmujących się regulacjami prawnymi, normalizacją i przepisami technicznymi w dziedzinie informatyki i telekomunikacji. Aktywna współpraca z EWG i dążenie do przyjęcia funkcji regionalnego centrum europejskiego mogą jednak doprowadzić do konfliktów z polityką USA w tej materii, związanych z wyraźnymi antagonizmami między EWG a USA na polu informatyki.

Wytyczaniu polskiej strategii informatycznej musi towarzyszyć ściśle i sprawne koordynowanie wszystkich międzynarodowych działań rządu w sprawach dotyczących informatyki i telekomunikacji. Jednym z istotnych zadań takiej koordynacji powinna być kontrola wszelkich publicznych przedsięwzięć informatycznych realizowanych w ramach pomocy zagranicznej dla Polski pod kątem ich zgodności z przyjętą strategią i polskimi normami.

Istotnym elementem strategii rządu powinna być odpowiednia polityka wobec fachowych kadr informatycznych. Dość dobry w chwili obecnej stan kadry informatycznej w Polsce wynika z uwarunkowań, które już nie istnieją. Zasoby kadrowe ulegają szybkiej erozji przez emigrację zarobkową na skutek chronicznego niedofinansowania uczelni akademickich i jednostek badawczych. Erozji ulega też kadra informatyczna w sektorze publicznym, nie wytrzymującym konkurencji z sektorem prywatnym. Jej miejsce zajmują często niedouczeni półamatorzy.

W celu powstrzymania tych niekorzystnych procesów rząd powinien podjąć następujące działania:

- wzorem wielu krajów rozwiniętych wprowadzić istotne dodatki płacowe dla kadry dydaktycznej w najlepszych instytutach informatyki i dla informatyków prowadzących poważne badania naukowe;

- doprowadzić do włączenia polskich zespołów do międzynarodowych programów badawczych (np. ESPRIT) według zasad przyjętych w EWG i finansować polską część tych projektów;

- wydzielić środki na długoterminowe zatrudnienie w Polsce kilku wybitnych specjalistów z wybranych działów informatyki i powierzyć im katedry w szkołach wyższych o największym dorobku;

- wydzielić środki na poprawę bazy dydaktycznej i badawczej w najlepszych instytutach krajowych;

- radykalnie podwyższyć wynagrodzenia informatyków zatrudnionych w sektorze publicznym, wcześniej poddając weryfikacji osoby już tam zatrudnione;

- wprowadzić regulacje prawne określające zakres uprawnień zawodowych i odpowiedzialności informatyków oraz zobiektywizowane kryteria kwalifikacji zawodowych. Wzorem powinny być zalecenia CEPIS (*Council of European Professional Information Processing Societies*) przygotowane dla EWG;

- wprowadzić system zachęt fiskalnych dla pracodawców gwarantujących zatrudnionym informatykom stałe szkolenie zawodowe prowadzone przez licencjonowane instytucje; licencjonowanie kursów szkoleniowych należy powierzyć stowarzyszeniu zawodowemu, np. Polskiemu Towarzystwu Informatycznemu, działającemu w porozumieniu z Ministerstwem Edukacji Narodowej;

- stworzyć warunki sprzyjające racjonalnej polityce wydawniczej w zakresie profesjonalnej informatyki.

Zaniechanie przedstawionych działań spowoduje w krótkim czasie konieczność angażowania w Polsce zagranicznych informatyków i znacznie zwiększy koszty rozwoju zastosowań informatyki.

Zrąb strategii informatycznej państwa powinny tworzyć przedsiębiorstwa o znaczeniu publicznym. Należą do nich: gospodarka publicznymi zasobami informacyjnymi, informatyzacja administracji państwowej i samorządowej oraz powiązana z nimi budowa sieci komputerowych. Głównym warunkiem powodzenia takich przedsięwzięć jest ich dobre przygotowanie

organizacyjne, a także ścisły nadzór nad ich realizacją ze strony zamawiającego. Doświadczenia wskazują, że większość przedsięwzięć była dotychczas źle przygotowana. Należy więc wprowadzić następujące reguły organizacji informatycznych przedsięwzięć publicznych:

- realizację systemu informatycznego powinno poprzedzać sformułowanie jego dokładnych założeń funkcjonalnych, przygotowanych przez zespół z udziałem wysoko kwalifikowanych informatyków;

- założenia funkcjonalne powinny być podstawą konkursu na wykonanie „systemu pod klucz” (tj. sprzęt i oprogramowanie razem);

- każde przedsięwzięcie powinno mieć tylko jednego głównego wykonawcę, odpowiedzialnego za jego realizację;

- duże przedsięwzięcia powinny być przedmiotem konkursów międzynarodowych;

- jednym z istotnych warunków konkursowych powinno być wymaganie angażowania polskich firm jako podwykonawców istotnych części systemu;

- rozstrzygnięcie konkursu powinno uwzględniać opinię zespołów niezależnych konsultantów;

- zamawiający powinien sprawować ścisły i bezpośredni nadzór nad przestrzeganiem przyjętych założeń i polskich norm;

- wszystkie publiczne systemy informatyczne powinny podlegać obowiązkowej atestacji;

- oferty dostarczenia systemów informatycznych w ramach pomocy zagranicznej powinny być rozważane pod kątem ich dostosowania do polskich norm standardów.

Przestrzeganie wymienionych reguł wiąże się z kosztami sięgającymi 10–12% wartości gotowych systemów. Zaniechanie tych nakładów musi nieuchronnie prowadzić do powstania systemów nie spełniających oczekiwań.

Ważnym celem strategicznym jest właściwa gospodarka zasobami informacyjnymi i rozwój tych zasobów. Należą do nich m.in. zbiory przepisów prawnych, wszelkie dane ewidencyjne, gospodarcze i statystyczne. Użytkowane dotychczas informatyczne systemy gospodarki zasobami informacyjnymi wymagają gruntownej i pilnej reorganizacji. Jej potrzeba wynika ze zmiany relacji państwo-obywatel oraz stąd, że dane gromadzone w niektórych takich systemach (jak np. PESEL) naruszają prawa osobiste obywateli. Inne systemy, tworzone lokalnie (np. ewidencja pojazdów mechanicznych), są wzajemnie niespójne. Budowa systemów nowych i reorganizacja starych powinny być skoordynowane z tworzeniem rozległych sieci komputerowych. Ciężar niezbędnych inwestycji – ze względu na ich strategiczny charakter, poufność danych i duże koszty realizacji systemów informacyjnych – musi wziąć na siebie państwo. Do najważniejszych zadań w tym zakresie należą:

- prawne ustalenie wymagań funkcjonalnych, jakie powinny spełniać powszechne systemy informacyjne, zasad dostępu do zawartych w nich danych i ich ochrony, z wykorzystaniem doświadczeń EWG;

- budowa dostępnego przez sieć komputerową systemu informatycznego z pełną informacją o ustroju prawnym państwa;

- dostosowanie zakresu informacji zbieranych przez GUS do norm EWG, połączone z prawnym uregulowaniem zasad dostępu do informacji gospodarczych i statystycznych przez podmioty prawne i obywateli oraz budowa systemu informatycznego dostarczającego takich informacji;

- ustalenie norm funkcjonalnych, jakie powinny spełniać gminne systemy informacyjne w odniesieniu do zadań zleczanych gminom przez administrację państwową; częściowe finansowanie gminnych systemów informacyjnych;

- pełna i jednolita informatyzacja systemu ewidencji gruntów i ksiąg wieczystych.

Ponadto rząd powinien wspierać działania tych podmiotów, które zdecydują się na samodzielne inwestowanie w rozwój publicznych systemów informacyjnych kraju za pomocą gwar-

ncji kredytowych i odpowiedniej polityki fiskalnej. Projekty poza rządowe w tej dziedzinie powinny być merytorycznie nadzorowane przez odpowiednie służby państwowe lub niezależne, upoważnione do tego instytucje.

Osobne pole działań strategicznych stanowi informatyzacja administracji państwowej. Obecny stan w tej dziedzinie jest bardzo niezadowolający. Organy i służby państwowe eksploatują wiele przestarzałych systemów, które z trudnością, jeśli w ogóle, poddadzą się niezbędnej modernizacji. Co więcej, systemy działające w różnych resortach, a niekiedy nawet w jednym, są wzajemnie niezgodne, co uniemożliwia wymianę informacji; dotyczy to także systemów nowo tworzonych. Poważnym problemem jest brak (z niewielkimi wyjątkami) fachowej kadry informatycznej. Niezbędna poprawa istniejącego stanu wymaga szybkiego podjęcia wielu działań, do których przede wszystkim należą:

- opracowanie założeń informatyzacji administracji państwowej na podstawie przyjętej strategii gospodarczej;
- wydzielenie grupy systemów informatycznych, które muszą być projektowane centralnie i dopiero po atestacji wdrażane w jednostkach lokalnych; w grupie tej powinny się znaleźć systemy ewidencji i ruchu ludności, podatkowe, obsługi socjalnej, ewidencji pojazdów mechanicznych oraz statystyki i analiz gospodarczych; inne systemy mogące być zamawiane i opracowywane lokalnie, z tym jednak, że przed rozpoczęciem eksploatacji powinny podlegać obowiązkowej atestacji;
- przestrzeganie przytoczonych zasad organizacji przedsięwzięć w projektowaniu, modernizacji i wdrażaniu systemów dla administracji państwowej;
- unifikacja systemów, oprogramowania i sprzętu komputerowego w drodze przyjęcia koncepcji systemów otwartych, charakteryzujących się dużą elastycznością;
- wyeliminowanie z użytkowania w administracji państwowej nie licencjonowanego oprogramowania;
- podjęcie prac nad ustawowym określeniem zakresu ważności prawnej dokumentów uzyskiwanych z systemów informatycznych;
- stworzenie ośrodka szkolenia pracowników administracji państwowej w zakresie podstaw użytkowania informatyki oraz pragmatyki jej stosowania.

Wbrew ogólnie przyjętemu poglądom, informatyzacja administracji państwowej nie przyniesie znacznych oszczędności. Płynące z niej korzyści ujawnią się dzięki uporządkowaniu struktur organizacyjnych zarządzania, poprawie obsługi klientów, unowocześnieniu instytucji państwowych i przyjęciu nowych metod podejmowania decyzji na podstawie pełniejszych informacji, a także dzięki dopasowaniu do już z informatyzowanych instytucji współpracujących.

Charakterystyczną cechą większości współczesnych systemów informatycznych jest ich rozproszenie, związane z zastosowaniami, które z natury rzeczy obejmują rozległy obszar geograficzny. Podstawą budowy takich systemów są sieci komputerowe, w tym przy zastosowaniach w skali państwowej – sieci rozległe, dotychczas w Polsce niedostępne. Konieczność budowy takich sieci nie ulegają wątpliwości. Podstawowe prace, jakie należy podjąć w tym kierunku, są następujące:

- zbudowanie, wdrożenie i rozpowszechnienie sieci publicznej z komutacją pakietów X.25 i stymulowanie rozwoju podobnych sieci prywatnych; budowana od długiego czasu sieć POLPAK nie spełnia wymagań powszechności dostępu, należy więc rozszerzyć jej koncepcję i zintensyfikować budowę;
- stymulowanie powstawania prywatnych, miejskich i między-miastowych, sieci kręgosłupowych z zastosowaniem technologii sieci lokalnych i sieci cyfrowych (kanały 2 Mbit/s) oraz uruchomienie programów badawczych dotyczących takich sieci, najlepiej we współpracy z EWG;
- przyjęcie jako polskich norm standardów międzynarodowych dotyczących sieci komputerowych oraz zapewnienie peł-

naprawnego udziału Polski w organizacjach opracowujących takie standardy;

- wprowadzenie obowiązku homologacji urządzeń i oprogramowania sieciowego;
- podjęcie starań zmierzających do integracji polskich sieci komputerowych z sieciami międzynarodowymi oraz do usunięcia ograniczeń w dostępie do danych i usług w sieciach;
- wprowadzenie własnych, oryginalnych rozwiązań ochrony danych w sieciach komputerowych.

Strategiczne działania państwa w dziedzinie informatyki muszą objąć również politykę gospodarczą wobec firm zajmujących się produkcją sprzętu komputerowego i oprogramowania oraz usługami informatycznymi. Polski przemysł komputerowy oraz przemysł produkcji elementów elektronicznych znajdujących zastosowanie w informatyce są w głębokim kryzysie, spowodowanym stosowaniem przestarzałych technologii i upadkiem RWPG. Równie poważny kryzys przechodzą przemysłowe instytuty badawcze. Firmy prywatne zajmują się głównie produkcją prostego oprogramowania aplikacyjnego, usługami informatycznymi, pośrednictwem w sprzedaży importowanego sprzętu komputerowego i oprogramowania oraz działalnością szkoleniową. Są one na ogół niedoinwestowane i pozbawione perspektyw rozwoju, mimo że zatrudniają dobrych fachowców. W Polsce istnieje też kilkanaście przedstawicielstw firm zagranicznych prowadzących działalność promocyjną. Z drugiej strony, światowy przemysł informatyczny, w tym zwłaszcza przemysł usług informatycznych, rozwija się bardzo dynamicznie, a nowy ustrój Polski daje nam szansę włączenia się do tego procesu. Porównanie skutków rozwoju informatyki w USA, gdzie są finansowane zastosowania, informatyką w Europie zachodniej, gdzie finansuje się przemysł informatyczny, wskazuje na wyższość finansowania zastosowań. W związku z tym uważamy, że należy przyjąć następujące zasady polityki gospodarczej:

- państwowe przedsiębiorstwa informatyczne powinny podlegać ogólnym regułom gry rynkowej; tylko taka strategia może wymusić na nich zmiany w kierunku powiązania z rynkiem światowym;
- wszystkie podmioty gospodarcze w sferze działalności informatycznej powinny być traktowane jednakowo;
- należy stymulować wyłącznie zastosowania informatyki, a zwłaszcza zastosowania łączące dostęp do światowej technologii z wykorzystaniem polskiego potencjału intelektualnego. Stymulowanie rynku zastosowań informatyki powinno być niezbędnym elementem strategii rządowej, ponieważ główny ciężar inwestowania w rozwój zastosowań muszą ponosić podmioty poza rządowe. Takie inwestowanie leży niewątpliwie w interesie państwa, jako jeden ze środków zmiany struktury gospodarki, przezwyciężenia kryzysu gospodarczego i dostosowania polskich instytucji do infrastruktury EWG. Najważniejszymi mechanizmami stymulacyjnymi, stosowanymi przez rząd wobec innych podmiotów, powinny być:
- zasadnicze skrócenie okresu amortyzacji informatycznych środków trwałych;
- generalne i trwałe zniesienie ceł na sprzęt informatyczny, podzespoły stosowane w komputerach, oprogramowanie i materiały informatyczne;
- podniesienie dolnej granicy wartości środków informatycznych, powyżej której są one traktowane jako środki trwałe i podlegają amortyzacji (ułatwi to zakup licencjonowanego oprogramowania);
- wyodrębnienie z puli kredytów przeznaczonych na restrukturyzację gospodarki części przeznaczanej na inwestycje informatyczne oraz częściowe gwarantowanie przez rząd spłaty takich kredytów;
- wprowadzenie ułatwień w leasingu sprzętu i oprogramowania komputerowego.

Proponowane mechanizmy powinny pośrednio poprawić stan polskich firm informatycznych, prowadząc do ich wzmocnienia i przyszłej skutecznej konkurencji na rynkach światowych.

Działania strategiczne powinny również objąć niezmiernie ważny problem standaryzacji sprzętu i oprogramowania informatycznego. Obecnie większość norm i standardów międzynarodowych nie ma swych odpowiedników w rejestrze Polskich Norm lub normy tam zawarte są przestarzałe. W związku z tym należy:

- zweryfikować polskie normy informatyczne pod względem ich aktualności i kompletności;

- szybko opracować nowe standardy i normy dotyczące sprzętu komputerowego, 8-bitowych kodów polskich znaków, układów klawiatur, protokołów komunikacyjnych i innych norm uzupełniających;

- zacieśnić i rozszerzyć współpracę z Międzynarodową Organizacją Standardów (ISO) i narodowymi organizacjami standaryzacyjnymi w celu uwzględnienia w przyszłych międzynarodowych normach polskich poglądów i wymagań; szczególnie ważne jest podjęcie współpracy z Czecho-Słowacją i Węgrami w sprawie kodów znaków narodowych;

- zainicjować prace na temat metod atestacji jakości i bezpieczeństwa oprogramowania oraz zasad eksploatacji systemów informatycznych;

- rozpocząć prace w celu zastosowania w gospodarce narodowej klasyfikacji gospodarczych EWG;

- wymagać dostarczania pełnych polskich wersji oprogramowania od wykonawców kontraktów na informatyzację administracji państwowej.

Rozwój zastosowań informacji wiąże się ze zjawiskami patologicznymi, które wymagają systematycznego przeciwdziałania ze strony państwa. Przepęstwa informatyczne można podzielić na dwie kategorie: przepęstwa popełnianie z użyciem środków informatyki oraz przepęstwa wobec systemów informatycznych. Podczas gdy pierwsza kategoria przepęstwa ma swe odpowiedniki klasyczne ujęte w Kodeksie Karnym, przepęstwa wobec systemów informatycznych wymagają specjalnych przepisów prawa karnego. Przeciwdziałanie przepęstwom z użyciem środków informatyki wymaga:

- przeszkolenia odpowiednich pracowników administracji państwowej i samorządowej oraz wymiaru sprawiedliwości;

- przygotowania zestawu przepisów ułatwiających informatyczną kontrolę legalności czynności wykonywanych z użyciem systemów informatycznych i wprowadzenie ich w życie w odniesieniu do publicznych systemów informatycznych oraz innych systemów z nimi powiązanych;

- podjęcia prac nad budową informatycznych narzędzi kontroli i inspekcji systemów informatycznych;

- powołania wyspecjalizowanych komórek w policji państwowej, Najwyższej Izbie Kontroli i Głównym Urzędzie Ciel do walki z przepęstwami informatycznymi;

- powołania niezależnej placówki atestacji oprogramowania pod kątem ułatwień kontroli i inspekcji oraz rzetelności. Przepęstwa wobec systemów informatycznych polegają na nieuprawnionej ingerencji w ich działanie, nieuprawnionym korzystaniu z ich zasobów, zniekształcaniu treści informacji, a także na niszczeniu systemów lub ich elementów. Przeciwdziałanie takim przepęstwom wymaga wielu przedsięwzięć; należą do nich:

- ustawienie normy ustalającej, że każdy akt nieupoważnionego wejścia do systemu komputerowego, modyfikacji programu lub danych jest czynem karalnym bez względu na intencje i skutki działania sprawcy;

- opracowanie zbioru wymagań względem bezpieczeństwa systemów, programów i danych informatycznych według zaleceń organizacji międzynarodowych (ISO, CCITT, EWG);

- ustawowe określenie stopni zabezpieczenia dla poszczególnych państwowych i publicznych systemów informatycznych,

a także dla innych systemów, których bezpieczeństwo ma bezpośredni wpływ na bezpieczeństwo i prawa obywateli;

- opracowanie zaleceń dotyczących przeciwwaryjnego bezpieczeństwa systemów i danych informatycznych;

- powołanie placówki atestacji bezpieczeństwa systemów informatycznych;

- powołanie wyspecjalizowanej komórki policji państwowej do zwalczania przestępstw wobec systemów informatycznych.

Sformułowaniu strategii informatycznej rządu powinny towarzyszyć odpowiednie prace legislacyjne. Główny obszar proponowanej regulacji ustawowej dotyczy systemów informatycznych o znaczeniu publicznym, zasad ich tworzenia, dopuszczania do eksploatacji oraz wykorzystywania gromadzonej i przetwarzanej informacji. Część zagadnień wymagających takich przedsięwzięć legislacyjnych tworzy nieuregulowaną dotychczas w Polsce sferę zasad gromadzenia informacji, ochrony danych i ochrony dóbr osobistych potencjalnie zagrożonych w wyniku eksploatacji systemów informatycznych. Inne zagadnienia mieszczą się w sferze regulacji prawnych dotyczących normalizacji i jakości. Penalizacja określonych czynów stanowi uznanie faktu, że specyfika systemu informatycznego stwarza nową jakość w społecznych i ekonomicznych skutkach czynów i wymaga specjalnych sankcji. Całościowa regulacja powyższych zagadnień oraz niezbędna nowelizacja wielu przepisów wymagają odrębnego przedsięwzięcia legislacyjnego, które powinno być przygotowane przez specjalnie powołany zespół ekspertów.

Jednoznaczne i szybkie rozstrzygnięcia legislacyjne wymaga kwestia ochrony prawnej oprogramowania, podnoszona od dawna przez Polskie Towarzystwo Informatyczne. Każde przedsięwzięcie w tym zakresie powinno uwzględniać zarówno opinię ekspertów-informatyków, jak i określoną politykę w stosunkach z granicą.



OŚRODEK POSTĘPU TECHNICZNEGO

ul. Bytowska 1b, 40-955 Katowice
tel. 596-061...7, faks 588-919
teleks 0312458 OPT PL

Zaprasza w dniach 6-7.05.1992 r.
na seminarium

Podstawowe problemy prawa komputerowego

W programie:

- * ochrona programów komputerowych w Polsce,
- * dyrektywa EWG z 1991 r.,
- * poza prawno-autorskie podstawy ochrony oprogramowania,
- * umowa licencyjna,
- * umowa o stworzenie programu,
- * umowa pomiędzy pracownikiem a pracodawcą.

Dyskusje, konsultacje indywidualne.

Udział w seminarium należy zgłosić
do dnia 15 marca 1992 r.

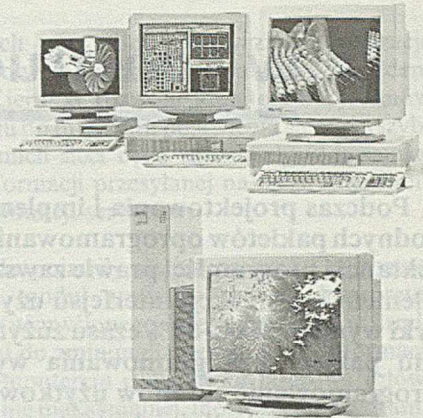
Informacje: BIURO ORGANIZACYJNE
Katowice,
tel. 596-061...7 wew. 278

01/192

Jeszcze raz liczby mówią za siebie.

6K

*) Zalecana cena za model 705 19" Greyscale, 8 MB,
Diskless wynosi US\$ 6.275,-.

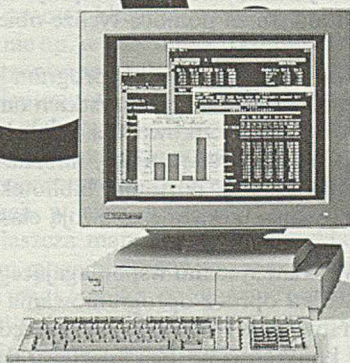


Nowe stacje robocze RISC marki HP
Apollo modele 705 i 710.

Dane liczbowe:

	Model 705	Model 710
MIPS	35	57,9
SPECS	34	49,7
MFLOPS	8	12,2

35



Czy mamy podać jeszcze więcej liczb?

Proszę do nas zadzwonić.

Hewlett-Packard Polska sp.z o.o.

Janusz Rybnik, Janusz Majchrzak

Ul. Nowelska 6

01447 Warszawa

Tel: 375065, Fax: 374783



**HEWLETT
PACKARD**

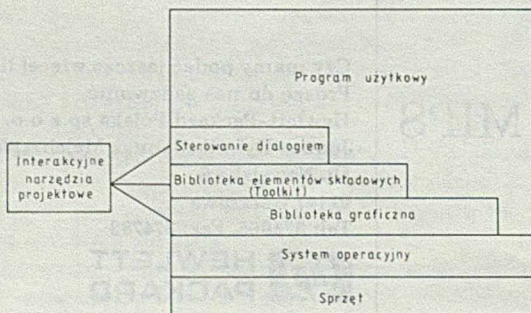
Nowe techniki w konstruowaniu interfejsów użytkownika

Podczas projektowania i implementowania różnorodnych pakietów oprogramowania użytkowego projektanci i programiści prawie zawsze stykają się z problemem konstrukcji interfejsu użytkownika. Z praktyki wynika, że 60–80% czasu zużytego na uruchomieniu pakietu oprogramowania wykorzystuje się na programowanie ekranów użytkowych, zwykle wykonywanych techniką ich ręcznego „zaszywania” w tekst źródłowy. Zastosowanie nowych, sformalizowanych technik konstrukcji interfejsów użytkownika otwiera szanse znacznego zmniejszenia kosztów wytwarzania oprogramowania.

W artykule omówiono typową architekturę systemu interfejsu użytkownika oraz dostępne produkty wspomagające programowania takich interfejsów.

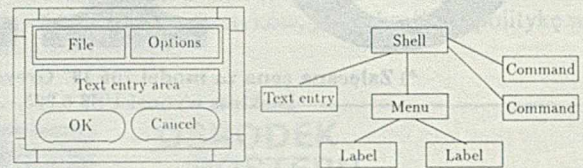
Struktura oprogramowania interfejsu użytkownika

Architekturę typowego systemu interfejsu graficznego przedstawiono na rys. 1. Biblioteka graficzna, zbudowana z wykorzystaniem możliwości sprzętu i systemu operacyjnego zawiera zestaw elementarnych operacji graficznych (procedury rysujące figury geometryczne) oraz jest odpowiedzialna za zarządzanie hierarchią okien, czyli wirtualnych urządzeń wyjściowych dla procedur geometrycznych. Zarządzanie oknami na poziomie biblioteki graficznej polega na uwzględnianiu przy rysowaniu wzajemnych relacji między oknami (przesłanianie), oraz na odpowiedniej obsłudze zmian tych relacji (np. przerysowywanie obszarów odsłoniętych po usunięciu zakrywającego okna). Do zadań tej warstwy należy też śledzenie stanu urządzeń wejściowych (myszy i klawiatury) oraz informowanie wyższych warstw oprogramowania o określonych zdarzeniach (np. wykrycie wejście kursora graficznego do obszaru pewnego okienka).



Rys. 1. Typowa architektura oprogramowania interfejsu użytkownika

Nad biblioteką graficzną jest zbudowana biblioteka gotowych elementów składowych interfejsu (ang. *widgets*). W jej skład wchodzi pewna liczba obiektów prostych (np. etykieta, przycisk funkcyjny, pole redakcyjne) oraz zestaw tzw. pojemników służących do wiązania obiektów prostych w potrzebne struktury (np. menu jest pojemnikiem zawierającym pewną liczbę przycisków funkcyjnych). Biblioteka ta (zwana też *toolkit*) jest pewną analogią do zestawu typów prostych i mechanizmów wiązania ich w struktury złożone w językach wysokiego poziomu typu C czy Pascal. Na rysunku 2 przedstawiono strukturę przykładowego obiektu dialogowego zbudowanego z elementów prostych i pojemników.



Rys. 2. Przykładowy obiekt dialogowy

Najwyższa warstwa (Sterowanie dialogiem) jest odpowiedzialna za sterowanie dialogiem, na który można spojrzeć jak na protokół komunikacji między dwoma procesami, przy czym jeden z nich jest operatorem komputera wprowadzającym ciąg akcji (ruchów myszki i naciśnięć klawiszy), a drugi procesem tłumaczącym akcje operatora na odpowiednie zachowania programu użytkowego. Z kolei program użytkowy modyfikuje stan obiektów dialogowych na ekranie, przysyłając informacje do operatora i powodując w ten sposób zainicjowanie jego akcji. O ile więc toolkit służył do tworzenia złożonych obiektów komunikacyjnych, o tyle warstwa sterowania dialogiem zarządza wymianą informacji między użytkownikiem a programem użytkowym za pomocą tychże obiektów.

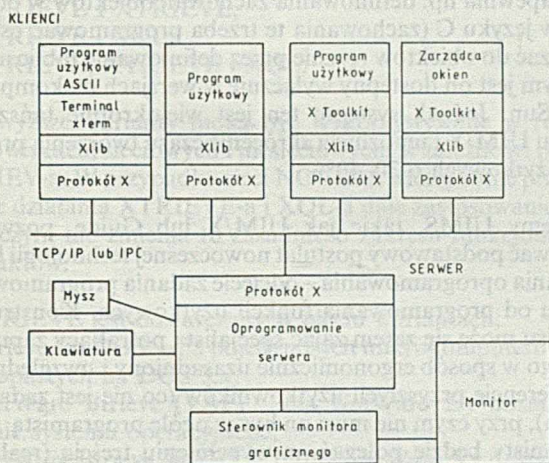
Jak widać na rys. 1 program użytkowy może korzystać bezpośrednio z usług wszystkich omówionych warstw. Dlatego te programy użytkowe, dla których możliwości obiektów toolkitu są niewystarczające, można częściowo lub w całości zrealizować na poziomie biblioteki graficznej – architektura warstwowa wykazuje tu swoją elastyczność.

W skład systemu wspomagającego tworzenie interfejsu użytkownika oprócz oprogramowania realizującego poszczególne warstwy muszą wejść jeszcze narzędzia służące do specyfikowania konkretnej postaci interfejsu na każdym z trzech rozważanych poziomów. Są to albo interaktywne programy typu CAD, albo kompilatory języków specyfikacji interfejsów.

System X Window

System X Window (w skrócie X) jest graficznym systemem okienkowym pracującym w środowisku sieciowym. Powstał on w 1984 r. w Massachusetts Institute of Technology, początkowo jako pakiet oprogramowania służącego głównie do celów dydaktycznych. W następnych latach systemu udoskonalono i rozbudowywano – ostatnia jego wersja nosi nazwę *X version 11* i jest w coraz większym stopniu uważana za przemysłowy standard oprogramowania graficznego. Oprogramowanie X Window jest w zasadzie dedykowane dla systemu UNIX, ale powstały też jego warianty dla innych systemów, np. dla VAX/VMS (pakiet *DecWindows*). X jest w swoich założeniach systemem otwartym o dużej elastyczności (duży stopień sparametryzowania), dzięki czemu stanowi odpowiednią podstawę do konstruowania zaawansowanych interfejsów użytkownika.

Systemem X Window jest zbudowany według zasad konstrukcji architektury typu klient-serwer. Serwerem (ang. *client-server*) jest proces implementowany w środowisku sieciowym jako specjalizowany komputer (X terminal) lub jako driver graficznego urządzenia wejścia-wyjścia, będący elementem oprogramowania stacji roboczej. Proces ten akceptuje określony zestaw poleceń graficznych (odpowiadający zestawowi procedur biblioteki graficznej Xlib [4]) i przejawia własną aktywność, polegającą na generowaniu zdarzeń będących wynikiem działalności operatora lub zmian stanu serwera. Klientem jest proces użytkowy dokonujący operacji wejścia-wyjścia za pośrednictwem serwera. Architektura systemu X Window przedstawiono na rys. 3.



Rys. 3. Architektura systemu X Window

Konstrukcja X Window umożliwia zarówno jednoznaczny komunikację pojedynczego procesu klienta z wieloma serwerami (co może mieć zastosowanie np. w programach komunikacyjnych w rodzaju poczty elektronicznej), jak i umieszczenie wielu interfejsów graficznych klientów w pojedynczym serwerze. Architektura taka umożliwia efektywne wykorzystanie systemu rozproszonego. Występują tu jednak analogiczne trudności jak w zagadnieniach obliczeń rozproszonych związane z relatywnie dużym kosztem komunikacji w sieci lokalnej. Duże ilości informacji graficznej (a zwłaszcza map bitowych) przesyłanej między klientami a serwerem mogą powodować znaczne opóźnienia w działaniu interfejsu użytkownika, co znacznie pogarsza komfort pracy. System X Window zawiera pewne rozwiązania usprawniające przesyłanie informacji przez sieć, takie jak np. konstrukcja protokołu komunikacji klient-serwer (X protocol [5]) oparta na buforowaniu i przesyłaniu informacji (poleceń

i zdarzeń) w większych porcjach; przesyłanie większych pakietów pozwala bowiem osiągnąć lepszy stosunek informacji użytecznej do informacji kontrolnej (nagłówki protokołów niższych warstw komunikacji).

Istotne z punktu widzenia efektywności jest też zastosowanie inteligentnego serwera, który na podstawie niewielkiej liczby kilkubajtowych poleceń potrafi skonstruować równoważny złożony obraz o rozmiarach rzędu megabajtów) we własnej pamięci, zamiast blokować sieć transmisją mapy bitowej.

Mimo zastosowanych uprawnień praktycznych realizacje systemów typu X Window nie są dobrze oceniane pod względem szybkości działania. Kierunki rozwoju architektury systemów graficznych tego rodzaju do momentu pojawienia się superszybkich (gigabitowych) tanich sieci będą zatem szły w kierunku minimalizacji ilości informacji przesyłanej na łączu klient-serwer oraz rozszerzania funkcji serwerów.

Pewne objawy trendu pojawiły się już na rynku w postaci komputerów, których moduł graficzny zawiera interpreter PostScriptu [3] – języka wymyślonego i zrealizowanego w firmie A dobre między innymi do zminimalizowania ilości informacji przesyłanej między komputerem a drukarką laserową (głównym celem było uzyskanie maksymalnej, możliwej do uzyskania na tym urządzeniu, jakości druku). Rozwiązanie to świetnie pasuje również do wyświetlaczy graficznych dużej rozdzielczości i zostało zastosowane np. w komputerze NeXT (Display PostScript). Karty PostScriptu zawierają również komputery firmy Sun Microsystems (pakiet XNeWS) oraz stacje robocze RS 6000 firmy IBM. Jak się wydaje, kolejnym krokiem będzie przeniesienie większości funkcji logicznego interfejsu użytkownika do serwera, który stanie się w ten sposób złożonym automatem odpowiedzialnym za większość operacji komunikacji z użytkownikiem.

Systemy specyfikowania interfejsów użytkowych

Z systemów dostępnych na rynku oprogramowania największe znaczenie zyskały dwa produkty zbliżone do siebie pod względem właściwości funkcjonalnych. Pierwszy z nich to system Motif firmowany przez *Open Software Foundation* (OSF) – organizację skupiającą kilkunastu znanych producentów oprogramowania i sprzętu, której działalność ma na celu tworzenie powszechnie akceptowanych standardów oprogramowania. Drugi z systemów – *Open Look* – należy do niekoronowanego króla rynku stacji roboczych, tj. firmy Sun Microsystems. W ciągu ostatniego roku *Open Look* i *Motif* podlegały ciągłej konfrontacji, która zakończy się chyba na korzyść tego ostatniego, po wypuszczeniu przez kilka firm implementacji *Motif* na komputery Sun.

System taki, jak *Motif* lub *Open Look*, składa się z czterech istotnych elementów. Pierwszym z nich jest biblioteka podstawowych obiektów (ang. *gadgets, widgets*), takich jak przełączniki, różne rodzaje menu, wskaźniki analogowe (ang. *scrollbars*) oraz różne obiekty złożone (obiekty będące pojemnikami innych obiektów). W przypadku *Motif* biblioteka składa się z ponad czterdziestu różnych obiektów.

Drugim elementem jest translator języka opisującego sposób składania obiektów prostych w kompletne interfejsy zastosowań. Język ten umożliwia specyfikowanie atrybutów obiektów (kolorów, tekstów umieszczonych na etykietach, wartości początkowych zmiennych stanu interfejsu), relacji między składowymi interfejsu (np. menu jest pojemnikiem zawierającym

pewną liczbę obiektów podstawowych, a kompletny interfejs składa się z tegoż menu oraz obszaru wprowadzania tekstu, zawartych wewnątrz otoczki programu użytkowego (ang. *shell widget*), spełniającej funkcje komunikacji z programem zarządzającym ekranem), a także sposoby tłumaczenia akcji użytkownika na wywołania funkcji programu użytkowego (ang. *translation management*).

Następny element omawianych pakietów to zarządca ekranu (ang. *window manager*). Jest to szczególnego rodzaju program użytkowy, którego zadaniem jest kontrola położenia i rozmiarów interfejsów zastosowań znajdujących się na ekranie serwera. Ciekawą właściwością zarządców ekranu jest możliwość ich współpracy z dowolnymi programami użytkowymi, niezależnie od tego, czy zostały one skonstruowane za pomocą Open Look lub Motif, czy też zostały napisane na niskim poziomie za pomocą tylko biblioteki Xlib. Wynika to z dokładnego zdefiniowania funkcji zarządzania ekranem w systemie X Window, na którym oparto zarówno Motif, jak i Open Look, a więc także wchodzące w ich skład programy *olwm* (*Open Look Window Manager*) i *mwm* (*Motif Window Manager*).

Ostatni, ale wcale nie najmniej ważny składnik pakietów Motif i Open Look jest zbiorem specyfikacji określających pożądany sposób konstruowania interfejsów za pomocą opisanych powyżej narzędzi. Zbiór ten (*Motif Style Guide* w przypadku Motif i *Open Look Style Guide* dla Open Look) ma na celu utrzymanie logicznej spójności interfejsów różnych programów użytkowych, co ma bardzo istotne znaczenie dla łatwości opanowywania nowych tego rodzaju programów przez użytkowników. Zarówno fundacja OSF, jak i firma Sun dokonuje ocen zgodności różnych pakietów użytkowych ze swoim podręcznikiem stylu i wydaje stosowane certyfikaty.

Reguły stylu oparte są na założeniu, że podobna logicznie funkcja w różnych programach musi być uaktywniana w ten sam sposób. Przykładem może być tu operacja usuwania fragmentu tekstu w edytorze tekstowym i operacja usuwania plików dyskowych za pomocą programu zarządcy plików. Obie funkcje wykonuje się przez zaznaczenie za pomocą myszyżądanego zbioru obiektów (liter lub ikon oznaczających pliki) i wybraniu z menu Edit funkcji Cut (prawie każdy program użytkowy ma zestaw funkcji służących do komunikowania się z tzw. buforem selekcji mogącym przechowywać obiekty dowolnego typu).

Zintegrowane systemy zarządzania interfejsem użytkownika (UIMS)

Pakiety takie, jak Motif lub Open Look, mimo swoich zalet nie wyczerpują możliwości usprawniania konstrukcji interfejsów użytkownika. Wymagają one nadal ręcznego tworzenia dużej ilości kodu (czy to w języku C, czy w językach opisu interfejsów UIL i XView), czyli dobrej znajomości programowania przez projektanta interfejsów.

Pakiety oprogramowania typu UIMS (ang. *User Interface Management Systems*) takie jak UIM/X firmy Visual Edge lub Open Windows Developer's Guide firmy Sun [1] oferują rozwiązania minimalizującego rozmiary kodu wpisywanego ręcznie. Są one zbudowane zgodnie z zasadą WYSIWYG (ang. *What You See Is What You Get*), tzn. interfejsy są konstruowane na ekranie za pomocą interakcyjnego edytora graficznego, który umożliwia składanie okienek z elementów wybieranych z zestawu obiektów bibliotecznych (będących w przypadku UIM/X kompletnym zestawem obiektów podstawowych pakietu Motif). Edytor ten realizuje pierwszy etap konstrukcji interfejsu, tzn. specyfikację jego wyglądu.

Drugim etapem jest specyfikacja zachowania się interfejsu. W systemie UIM/X dokonuje się tego za pomocą edytora właściwości obiektów (ang. *property editor*), który umożliwia nadanie obiektowi właściwości statycznych, takich jak rozmiary, kolory itp., jak też właściwości specyfikujących zachowanie – czyli po prostu fragmentów kodu w języku C specyfikujących obsługę zdarzeń dotyczących obiektu. Ciekawą własnością pakietu UIM/X jest wbudowany interpreter języka C, bardzo przydatny przy testowaniu interfejsów, będących następnym etapem sesji z systemem UIM/X. Testowanie polega po prostu na przełączeniu pakietu w tryb uaktywniający utworzone do tej pory obiekty wchodzące w skład programu użytkowego, które reagują na akcje użytkownika tak, jak podczas normalnej sesji. Testowanie umożliwia ocenę poprawności działania i wygody użytkownika programu, a także wykonanie poprawek (za pomocą edytorów wyglądu i własności) bez konieczności wykonywania jakichkolwiek kompilacji. Uzyskuje się w ten sposób ogromne przyspieszenie procesu tworzenia oprogramowania użytkowego.

Kolejne etapy budowy interfejsu to zapisanie jego specyfikacji na dysku oraz wygenerowanie kodu źródłowego (programów w języku C, specyfikacji UIL oraz plików konstrukcyjnych do kompilacji *makefile*), umożliwiającego uruchomienie programu działającego samodzielnie (bez środowiska uruchomieniowego UIM/X). Wygenerowany kod jest przenośnym dialektem języka C możliwym do skomplikowania na różnych realizacjach systemu UNIX, co umożliwia produkowanie oprogramowania dla różnych komputerów docelowych.

System Guide firmy Sun ma znacznie mniejsze możliwości – nie zapewnia np. definiowania zachowań obiektów w postaci kodu w języku C (zachowania te trzeba programować osobno i dołączać do obiektów ręcznie przez definiowanie tablic akcji). Poza tym jest on dostępny wyłącznie w wersjach na komputery firmy Sun. Jednak system ten jest wielokrotnie tańszy od systemu UIM/X i umożliwia skrócenie czasu tworzenia programów użytkowych o 30–40%.

Systemy UIMS, takie jak UIM/X lub Guide, pozwalają realizować podstawowy postulat nowoczesnej technologii konstruowania oprogramowania – odcięcie zadania programowania sprzęgu od programowania funkcji użytkowych. Konstrukcja interfejsu może się zatem zająć specjalista potrafiący zaprojektować go w sposób ergonomicznie uzasadniony i uwzględniający preferencje przyszłych użytkowników (co nie jest zadaniem łatwym), przy czym nie musi on być w ogóle programistą. Rola programisty będzie polegać na wypełnieniu treścią (realizacji funkcji użytkowych) obiektów stworzonych przez projektanta interfejsu. Efektem powinny być programy użytkowe dobrej jakości zarówno pod względem interfejsu, jak i wykonywanych funkcji, wytworzone w dość krótkim czasie.

Kierunki rozwojowe zintegrowanych systemów tworzenia programów użytkowych

Wydaje się, że korzyści uzyskane przez zastosowanie interakcyjnych technik konstruowania interfejsu użytkownika i generowanie kodu źródłowego mogą zostać znacznie zwiększone przez rozszerzenie możliwości obiektów bibliotecznych o pewne standardowe funkcje użytkowe (np. obiekt realizujący operacje przeglądania katalogu dyskowego). Obiekty te byłyby standardowymi serwerami usług często wykorzystywanych podczas tworzenia oprogramowania użytkowego.

dokończenie na s. 23

Od BTRIEVE-a do SQL-a

Nowoczesne, wielopoziomowe środowisko baz danych firmy NOVELL

Celem artykułu jest przedstawienie środowiska baz danych firmy NOVELL, a zwłaszcza podstawowych cech produktów tworzących to środowisko oraz wniosków praktycznego ich wykorzystywania.

Środowisko baz danych firmy NOVELL

Na środowisko baz danych firmy NOVELL składają się cztery produkty: BTRIEVE, XTRIEVE, XQL, SQL. Mogą być one wykorzystywane w następujących wariantach:

- sam BTRIEVE,
- BTRIEVE i XTRIEVE,
- BTRIEVE i XQL,
- BTRIEVE, XTRIEVE i XQL.

Powyższe warianty mogą być wykorzystywane w różnych środowiskach, sieciowych i niesieciowych, w zależności od typu BTRIEVE-a. W przypadku sieci NOVELL wielokrotne przyspieszenie działania XTRIEVE-a i XQL-a daje zastosowanie SQL, przy czym nie zmienia to zasadniczo zakresu funkcjonalnego produktów.

BTRIEVE jest dostarczany w pięciu wariantach:

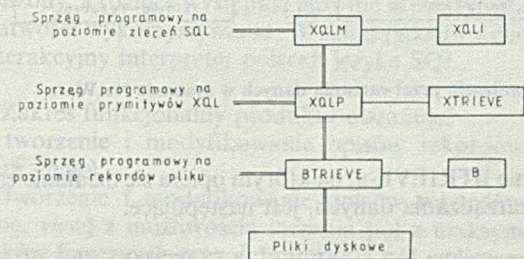
- Btrieve for DOS – dla pojedynczych mikrokomputerów oraz sieci opartych na DOS 3.x,
- NetWare Btrieve (VAP) – dla NetWare 2.x dostarczany w cenie systemu operacyjnego,
- NetWare Btrieve 386(NLM) – dla NetWare 3.x dostarczany w cenie systemu operacyjnego,
- Btrieve for OS/2,
- Btrieve for Windows.

XTRIEVE i XQL są dostarczane w dwóch wersjach: dla DOS i dla OS/2, przy czym wersja DOS współpracuje ze wszystkimi rodzajami BTRIEVE-a funkcjonującymi w środowisku dosowskim. SQL jest dostarczany w wersji NetWare VAP i w wersji NetWare 386 NLM.

Należy podkreślić, że wszystkie wersje produktów są funkcjonalnie zgodne, to znaczy programy napisane z wykorzystaniem odwołań do nich będą działały prawidłowo we wszystkich środowiskach dosowskich, bez żadnych zmian lub konieczności rekompilacji, pod warunkiem zainstalowania odpowiednich wersji produktów.

Pełny zakres wykorzystania przedstawianych produktów umożliwia dostęp do bazy danych (bezpośredni i za pomocą sprzęgów programowych) na trzech poziomach (rys. 1).

W przypadku zastosowania SQL-a, XQLP i XQLM są zastępowane produktem NSSERVER rezydującym na serwerze plików (ang. *file server*).



Rys. 1. Struktura środowiska baz danych firmy NOVELL

Na najniższym poziomie są dostępne operacje na rekordach plików, wchodzących w skład bazy danych, za pomocą sprzęgów programowych (Basic, C, Cobol, Pascal, Assembler) oraz dostęp interakcyjny, za pomocą programu pomocniczego B (ang. *utility*). Zarówno sprzęgi programowe, jak i program pomocniczy stanowią integralną część pakietu BTRIEVE.

Na poziomie średnim są dostępne operacje wykonywane za pomocą tak zwanych prymitywów XQLP. Są to funkcje umożliwiające tworzenie opisu bazy danych, modyfikowanie zawartości oraz wyszukiwanie odpowiednich informacji i przedstawienie jej w zadanych układach.

Wydzielenie tego poziomu było podyktowane chęcią zwiększenia efektywności przetwarzania przy praktycznym zachowaniu możliwości poleceń języka SQL, choć w bardziej skomplikowanej formie. Na poziomie tym są dostępne ww. sprzęgi programowe oraz program użytkowy XTRIEVE umożliwiający interakcyjne wykonywanie wszystkich operacji na bazie danych. Na poziomie najwyższym jest dostępne wykonywanie poleceń języka SQL, poprzez omawiane sprzęgi programowe oraz interakcyjny interpreter poleceń języka SQL o nazwie XQLI. W skład pakietu XTRIEVE wchodzi również moduł XQLP.

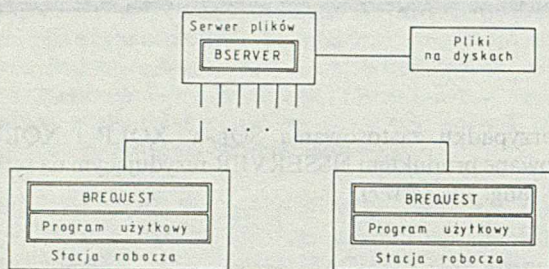
W skład pakietu XQL wchodzi moduły XQLP i XQLM, sprzęgi programowe drugiego i trzeciego poziomu oraz interpreter XQLI.

Pakiet SQL zastępuje jedynie moduły XQLP i XQLM oraz nie zawiera sprzęgów programowych ani modułów interakcyj-

nego dostępu i dlatego nie może być wykorzystywany samodzielnie, a jedynie łącznie z XQL-em lub (i) XTRIEVE-em.

Współdziałanie z siecią NOVELL

Bazy danych oparte na produktach firmy NOVELL działają efektywnie w różnych środowiskach systemowych i konfiguracyjnych. Szczególną jednak efektywnością wyróżniają się one przy współpracy z sieciowymi systemami operacyjnymi NetWare. Efektywność ta jest zdecydowanie większa niż efektywność innych podobnych produktów, zwłaszcza w przypadku dużych plików. Wynika to między innymi z przyjętej organizacji przetwarzania, zilustrowanej na rys. 2.



Rys. 2. Organizacja przetwarzania danych w systemie NetWare

Działanie BTRIEVE-a, na którym opiera się działanie całego systemu zarządzania danymi, jest następujące:

- program użytkowy wywołuje funkcję BTRV z odpowiednimi parametrami,
- wywołanie jest obsługiwane przez program BREQUEST rezydujący w stacji roboczej; program ten wysyła odpowiednie żądanie do serwera plików,
- proces BSERVER działający w serwerze plików odbiera żądanie programu BREQUEST, wykonuje odpowiednie operacje i zwraca wynik do stacji roboczej,
- BREQUEST w stacji roboczej otrzymuje odpowiedź serwera plików i zwraca ją w postaci odpowiednich parametrów do programu użytkowego,
- program użytkowy otrzymuje wynik operacji w odpowiednich parametrachwołania i wykorzystuje go w dalszej działalności.

Taka organizacja przetwarzania powoduje, że większość czynności wykonuje się bezpośrednio w serwerze plików, a sieć jest wykorzystywana tylko dwukrotnie: przy przesyłaniu żądania i przy otrzymywaniu rezultatu. Ma to szczególne znaczenie w przypadku wersji 5.10 BTRIEVE-a, która umożliwia jednorazowe przesyłanie wielu rekordów danych z możliwością określenia warunków selekcji. W tej sytuacji liczba przesyłań sieci jest wielokrotnie mniejsza niż w innych systemach baz danych.

W przypadku zastosowania SQL-a, w serwerze plików jest instalowany dodatkowy proces typu VAP (lub NLM) o nazwie NSSERVER, a w stacji roboczej – program rezydentny NSREQ. Odwołania do prymitywów XQLP lub w programie użytkowym bądź przy dostępie interaktywnym są realizowane w tym przypadku w analogiczny sposób jak polecenia BTRIEVE-a.

Taka organizacja przetwarzania powoduje, że praktycznie całe przetwarzanie jest wykonywane w serwerze plików, a sieć jest wykorzystywana tylko dwukrotnie: przy przesyłaniu żąda-

nia i przy otrzymywaniu rezultatu. Te dwa przesłania zastępują wiele, czasem dziesiątki tysięcy, przesyłań, jakie musiałyby być wykonane przy każdej innej organizacji przetwarzania.

W praktyce, przy operowaniu dużymi plikami (kilkadziesiąt tysięcy rekordów), zastosowanie SQL-a może dać nawet dziesięciokrotne przyspieszenie wykonywania poleceń języka SQL lub działania XTRIEVE-a.

BTRIEVE

BTRIEVE jest systemem zarządzania rekordami plików, umożliwiającymi tworzenie relacyjnych baz danych. System ten umożliwia tworzenie plików, których wielkość jest ograniczona praktycznie tylko pojemnością dysków. Można dzielić plik na kilka urządzeń dyskowych. Klucz indeksowy może składać się z kilku segmentów, to jest pól o dowolnej lokalizacji w rekordzie i dowolnego rodzaju. Łączna liczba segmentów kluczy indeksowych (ograniczająca samą liczbę kluczy) wynosi 24. Może to być w pewnych przypadkach dość istotnym ograniczeniem. Zarówno dane, jak i ścieżki dostępu są przechowywane w jednym pliku, przy czym organizacja tego pliku zapewnia dość dużą oszczędność pamięci w porównaniu do innych organizacji. Istnieje także możliwość tworzenia kluczy dodatkowych (ang. *supplemental key*), które mogą być deklarowane, tworzone i usuwane w dowolnym momencie istnienia pliku.

Indeks pliku może być modyfikowalny lub niemodyfikowalny, powtarzalny lub unikalny, zorganizowany według wartości rosnących bądź malejących. Przez deklarowanie kluczy pustych (ang. *null key*) lub kluczy ręcznych (ang. *manual key*) można uwzględnić w danej ścieżce dostęp tylko do niektórych, istotnych rekordów. Zmniejsza to zajętość pamięci dyskowej i poprawia efektywność odczytu.

Pliki tworzenia przy użyciu BTRIEVE-a, składają się ze stron. Długość strony, deklarowana przy tworzeniu pliku, może być wielokrotnością liczby 512, maksymalnie 4096 bajtów. Pierwsza strona pliku jest stroną nagłówkową, zawierającą informacje o pliku. Kolejne strony mogą zawierać rekordy pliku (ang. *data pages*) lub ścieżki dostępu (ang. *index pages*). Organizacja indeksu jest organizacją typu zrównoważonego B-drzewa.

Przy dopisywaniu rekordów do pliku, jego wielkość jest automatycznie zwiększana. Po wyczerpaniu dostępnego obszaru na danym dysku, można kontynuować plik na innym dysku. Przy usuwaniu rekordów z pliku, zwalniana przestrzeń dyskowa jest ponownie wykorzystywana w trakcie zapisywania nowych rekordów. Rekordy pliku mogą być stałej bądź zmiennej długości. Aby zmniejszyć zajętość dysku, można stosować następujące mechanizmy:

- obcinanie (ang. *truncation*) końcowych spacji w rekordach zmiennej długości,
- kompresję (ang. *compression*) powtarzających się bajtów.

Dostęp do kolejnych rekordów pliku można uzyskać:

- według fizycznej kolejności w pliku; jest możliwy odczyt rekordu o zadanym adresie fizycznym lub rekordu, pierwszego, ostatniego, następnego bądź poprzedniego;
- według wartości wybranego klucza dostępu; jest możliwy odczyt według warunków =, >, <, nie większy i nie mniejszy, a także odczyt pierwszego, ostatniego, następnego lub poprzedniego rekordu według wybranego klucza.

BTRIEVE udostępnia następujące metody ochrony spójności danych na wypadek awarii lub niewłaściwej pracy komputera:

- dla pojedynczego pliku ochroną kompletności zapisu, pole-

gającą na zapewnieniu zmodyfikowania bądź wszystkich niezbędnych elementów (rekordu, ścieżek dostępu, parametrów pliku), bądź żadnego z nich. Jeśli plik jest objęty mechanizmem TTS systemu NOVELL NetWare, to wykorzystuje się ten mechanizm, w przeciwnym razie wkorzystuje się własny mechanizm, zwany „pre-imaging”;

- dla grupy plików wykorzystuje się metodę ochrony transakcji. Początek transakcji jest zaznaczony instrukcją BEGIN TRANSACTION, a koniec transakcji – instrukcją END TRANSACTION. Jeżeli transakcja została rozpoczęta, lecz nie zakończona, np. ze względu na awarię systemu, zanik zasilania itp., to zostanie odtworzony automatycznie stan wszystkich plików taki, jaki był w momencie rozpoczynania transakcji. Również wykonanie instrukcji ABORT TRANSACTION (np. w przypadku stwierdzenia, że status wykonania zapisu w obrębie transakcji jest nieprawidłowy) powoduje przywrócenie stanu wszystkich plików z początku transakcji. Tak więc mechanizm transakcji powoduje, że albo wszystkie modyfikacje objęte transakcją zostaną zapisane prawidłowo, albo nie zostanie zapisana żadna z nich. Należy zaznaczyć, że transakcja blokuje wszystkie pliki, na których operacje są wewnątrz niej wykonywane, należy więc używać jej rozważnie, a w szczególności nie wykonywać wewnątrz niej odczytywania ekranu, połączonego z czekaniem na reakcję użytkownika;

- ochronę przed trwałymi uszkodzeniami pliku (dostępna w wersji 5.10), polegająca na zapisywaniu w odrębnym pliku wszystkich zmian w zawartości chronionego pliku, jakie zostały wykonane od momentu wykonania ostatniego BACKUP-u. W przypadku trwałego uszkodzenia chronionego pliku, jest możliwe odtworzenie jego **aktualnego** stanu, przez odczytanie wersji backup-owej i dopisanie do niej wszystkich modyfikacji dokonanych od czasu utworzenia tej wersji. Ta ostatnia operacja jest wykonywana przy użyciu programu pomocniczego (ang. *utility*) BROLLFWD.

Zapewnienie poprawnej wielodostępności plików jest realizowane przez:

- blokowanie rekordu przy użyciu instrukcji LOCK; opcje WAIT i NOWAIT specyfikują, czy zadanie, które usiłuje odczytać zablokowany rekord, ma czekać na jego odblokowanie, czy też zakończyć niepomyślnie operację;
- blokowanie plików przy użyciu transakcji;
- blokowanie na poziomie modyfikacji za pomocą biernego współdzielenia (ang. *passive concurrency*); jeśli rekord został zmieniony (przez innego użytkownika) między jego odczytaniem a próbą modyfikacji, to modyfikacja nie jest wykonywana i odpowiedni kod powrotu informuje o powstałej sytuacji; jest to najlepszy (z punktu widzenia użytkownika) sposób rozwiązania problemu wielodostępności plików, gdyż nie blokuje dostępu do żadnego rekordu, a jednocześnie zapewnia prawidłowe współdzielenie plików;
- automatyczne (przez odpowiedni kod powrotu) wykrywanie zakleszczeń, czyli wzajemnego zablokowania się użytkowników (ang. *deadlock*).

Ochrona danych przed nieuprawnionym dostępem jest realizowana przez:

- użycie normalnych mechanizmów systemu NOVELL NetWare, takich jak: hasła dostępu, prawa dostępu do plików, ograniczenia wykonywalności operacji;
- użycie hasła dostępu do pliku; hasło to musi być podawane (jako parametr wywołania) przy otwieraniu danego pliku; hasło może być wymagane tylko przy otwieraniu pliku w trybie umożliwiającym modyfikowanie jego zawartości, lub w każdym trybie otwarcia pliku;
- możliwość szyfrowania danych, przy czym jako klucz szyfro-

wania jest używane hasło dostępu do pliku (maksymalnie 8 znaków);

- otwieranie pliku w trybie wyłączności (ang. *exclusive model*), uniemożliwiającym dostęp do pliku innym użytkownikom.

BTRIEVE umożliwia również stosowanie, przy określaniu kolejności indeksów, innego alfabetu niż kod ASCII. Daje to m.in. możliwość porządkowania indeksów zgodnie z zasadami języka polskiego lub utożsamiania małych liter z dużymi.

XQL

XQL jest wielopoziomowym systemem tworzenia relacyjnych baz danych opartym na pakiecie BTRIEVE. Umożliwia rozdzielanie fizycznych struktur rekordów od struktur logicznych, wykorzystywanych w programach użytkowych. Sprzęgi programowe umożliwiają wykorzystywanie tego narzędzia przez programy użytkowe napisane praktycznie we wszystkich rozpowszechnionych dialektach języków programowania, dostępnych na mikrokomputerach. Sprzęgi te umożliwiają operowanie danymi na dwóch poziomach: poleceń języka SQL oraz tak zwanych prymitywów XQLP. Zakres funkcjonalny obu poziomów jest zbliżony, a różnice występują głównie w efektywności działania i łatwości wykorzystywania. W skład pakietu wchodzi również interakcyjny interpreter poleceń języka SQL.

Zakres funkcjonalny produktu obejmuje:

- tworzenie i modyfikowanie opisów rekordów fizycznych (ang. *table*),
- tworzenie i modyfikowanie opisów rekordów logicznych (ang. *view*) z możliwością łączenia pól z maksymalnie ośmiu plików fizycznych oraz definiowania pól wyliczanych,
- definiowanie praw dostępu poszczególnych użytkowników na poziomie rekordów i pól,
- wyszukiwanie informacji według zadanych kryteriów i układów sortowania oraz według wartości narastających lub malejących,
- uzupełnianie, modyfikowanie i usuwanie danych z bazy danych.

Słownik bazy danych składa się z siedmiu plików, zapisanych w standardzie BTRIEVE, zawierających: katalog plików, katalog pól, katalog ścieżek dostępu do plików, katalog atrybutów pól, katalog rekordów logicznych, katalog użytkowników, katalog praw dostępu.

XQL wykorzystuje i udostępnia wszystkie możliwości pakietu BTRIEVE w szczególności ochronę na poziomie transakcji, szyfrowanie itd. W zakresie ochrony przed niepowołanym dostępem, pakiet umożliwia definiowanie użytkowników z hasłami dostępu i grup użytkowników oraz nadawania im praw dostępu do plików i pól. Są to prawa odczytania i modyfikowania zawartości konkretnego pola lub prawa odczytania, modyfikowania, dopisywania i usuwania rekordów pliku oraz tworzenia plików.

Przy definiowaniu kryteriów selekcji system dokonuje optymalizacji metody wyszukiwania informacji przez dopasowanie kryteriów do istniejących w pliku kluczy indeksowych. Optymalizacja ta polega między innymi na wybraniu najbardziej odpowiedniego klucza dostępu. Optymalizacja nie działa, jeżeli jest wymuszone uporządkowanie wyszukiwanych danych. W przypadku uporządkowania niezgodnego z żadnym z kluczy dostępu, jest tworzony dodatkowy, zewnętrzny klucz dostępu, usuwany po wykonaniu polecenia. Operacja ta, zwłaszcza w przypadku dużych (zawierających kilkadziesiąt tysięcy rekordów) plików może być czasochłonna (trwa kilkanaście minut).

Zakres funkcjonalny języka SQL jest zgodny z ogólnie przyjętymi standardami tego języka, a w niektórych przypadkach stanowi ich rozszerzenie. Oprócz wcześniej wymienionych obejmuje on, również między innymi następujące możliwości:

- zagnieżdżanie instrukcji SELECT języka SQL,
- definiowanie alternatyw instrukcji SELECT (UNION),
- definiowanie podsumowań częściowych (GROUP BY) z operatorami: ilość, suma, wartość maskymalna, wartość minimalna, wartość średnia,
- łączenie do ośmiu plików fizycznych według kryteriów: równy, większy, mniejszy oraz produktu kartezjańskiego.

XTRIEVE

XTRIEVE jest interakcyjnym systemem zarządzania bazą danych opartym na poziomie prymitywów XQL. Ma wszystkie możliwości udostępniane przez BTRIEVE-a XQL-a, a w szczególności:

- definiowania i obsługi katalogów bazy danych,
- definiowania zakresu, postaci i formy prezentacji danych na ekranie monitora bądź na wydruku,
- definiowania wielocłonowych kryteriów selekcji danych,
- modyfikowania, usuwania i uzupełniania danych,
- definiowania praw dostępu do rekordów i pól,
- definiowania podsumowań częściowych i globalnych,
- łączenia do ośmiu plików fizycznych,
- wykorzystywania prostego generatora raportów.

XTRIEVE jest zorganizowany w formie wielopoziomowego systemu menu, a wszystkie operacje są wykonywane przez wybór jednej z proponowanych możliwości. Taka organizacja umożliwia łatwe wykorzystywanie nawet przez osoby nie znające pakietu i jego możliwości. Stanowi wygodne narzędzie dla osób nadzorujących pracę wielodostępnych baz danych, może być także wykorzystywany jako samodzielny system zarządzania bazą danych oraz do generowania prostych aplikacji.

SQL

SQL jest implementacją XQL-a (a dokładniej XQLP i XQLM) dla systemów funkcjonujących w sieci NOVELL. Działa jako proces VAP (NetWare 2.x) lub NLM (NetWare 3.x). Daje możliwość efektywnego operowania bazą danych zorganizowaną w jednym bądź w wielu serwerach plików.

W skład pakietu wchodzi: proces rezydujący w serwerze plików, NSREQ instalowany w stacjach roboczych, program instalacyjny oraz kilka poleceń konsoli serwera plików, umożliwiających kontrolowanie procesu rezydującego w serwerze plików. Pakiet nie zawiera sprzęgów programowych ani interpretera poleceń języka SQL, dlatego może być używany tylko łącznie z XQL-em (lub) XTRIEVE-em, bądź aplikacjami zawierającymi sprzęgi programowe do SQL-a.

Przykład praktycznego zastosowania

Z użyciem omówionych produktów wykonano i wdrożono system ewidencji materiałów w magazynach Zakładów Kineskopowych „Polkolor SA”. System funkcjonuje w sieci NOVELL o rozpiętości ponad 3 km i łącznej liczbie 34 stacji roboczych. Został wykonany z użyciem najnowszej wersji kompilatora COBOL/2 i pakietu wspomagającego WORKBANCH firmy MICRO FOCUS. Pakiet ten udostępnia zintegrowane środowisko programowania, umożliwiające wygodne redagowanie, kompilację, testowanie (animację) i konsolidację programów. Pakiet umożliwia m.in. wygodne projektowanie formatek ekranowych i automatyczne generowanie odpowiedniego kodu źródłowego, a także wygodne operowanie oknami programowymi (ang. *panels*). Możliwość wykorzystywania rozrzuconej pamięci operacyjnej komputera oraz wieloproceso-

wość pakietu w radykalny sposób przyspieszają proces kompilacji i konsolidacji dużych programów użytkowych.

W systemie *gospodarka materiałowa* wykorzystuje się wszystkie możliwości BTRIEVE-a oraz XTRIEVE-a i XQLI do kontrolowania danych i wykonywania nietypowych analiz i zestawień. System funkcjonuje w trybie on-line, to znaczy dane są wprowadzane bezpośrednio w magazynach, a informacje o materiałach są natychmiast aktualizowane i dostępne dla wszystkich użytkowników sieci. System udostępnia informacje o stanach i obrotach materiałów za dowolny okres w skali roku, a także umożliwia wykonanie różnych zapasów materiałowych.

Blisko roczna eksploatacja systemu upoważnia do wystawienia wysokiej oceny zastosowanym w nim rozwiązaniom. System jest całkowicie odporny na zakłócenia sieci, nie zanotowano przypadków naruszenia integralności danych lub uszkodzenia plików, mimo licznych problemów z siecią. System jest wygodny dla użytkowników i bezpieczny w eksploatacji, o czym świadczy bezproblemowa jego eksploatacja, mimo że jest wykorzystywany przez osoby bez przygotowania informatycznego, a często także o niskim poziomie wiedzy ogólnej. Jednocześnie, dzięki wykorzystaniu XTRIEVE-a i XQLI- system umożliwia szybkie wyszukiwanie informacji w dowolnym przekroju.

Planuje się dalsze rozwijanie systemu, zarówno w zakresie funkcjonalnym, jak i pełniejszego wykorzystywania sprzęgów wyższych poziomów w programach użytkowych.

Przedstawione produkty programowe firmy NOVELL tworzą środowisko relacyjnych baz danych spełniające wszystkie wymagania, stawiane przed pakietami baz danych. Środowisko to jest wewnętrznie spójne. Opisy rekordów fizycznych i logicznych, stworzone przy użyciu XTRIEVE-a, mogą być wykorzystywane w instrukcjach języka SQL i vice versa. Możliwy jest jednoczesny dostęp do tych samych plików fizycznych z dowolnego z omówionych poziomów. W programach aplikacyjnych można w dowolny sposób mieszać sprzęgi programowe z dowolnego poziomu. Wreszcie programy aplikacyjne, wykorzystujące sprzęgi programowe, mogą współdziałać z programami pomocniczymi (ang. *utility*), takimi jak XTRIEVE i XQLM.

Omawiane środowisko jest połączeniem dwóch światów, standardowego sposobu przetwarzania danych za pomocą programów, napisanych w językach trzeciej generacji, wykonujących standardowe operacje na rekordach plików fizycznych, oraz świata baz danych ze wszystkimi jego możliwościami. Możliwość łączenia tych dwóch światów w jednej aplikacji jest unikalną własnością tego środowiska.

Produkty firmy NOVELL są dostosowane do możliwości sprzętu mikrokomputerowego, a w szczególności do systemu operacyjnego NetWare. Efektem tego jest ich duża efektywność, która zwłaszcza w przypadku dużych baz danych (zawierających kilkadziesiąt czy kilkaset tysięcy rekordów), jest – jak wykazują testy praktyczne – zdecydowanie większa niż efektywność innych baz danych dostępnych na mikrokomputerach. Niska cena (łączny koszt zakupu wszystkich czterech omówionych produktów wynosi 2600 \$) powoduje, że produkty wydają się wręcz idealne dla wielu naszych zastosowań. Brak jakiegokolwiek reklamy ze strony firmy NOVELL i jej dealerów jest przyczyną małego ich rozpowszechnienia na polskim rynku.

Wnioski płynące z praktycznego wykorzystywania tych produktów upoważniają do twierdzenia, że sieci mikrokomputerowe oparte na produktach firmy NOVELL mogą z powodzeniem zastępować przetwarzanie na dużych komputerach, i to w dużo większej skali niż się dotychczas wydawało.

JUNISOFTEX Sp. z o.o.

44-100 GLIWICE ul. Konstytucji 11,
tel.-faks 31-75-10, 31-90-81 do 88 w. 250, 272, 282, teleks 036233

JUNISOFTEX – to firma z tradycjami i najdłuższymi doświadczeniami w eksploatacji wielodostępnych systemów komputerowych Novell w kraju.

JUNISOFTEX – to najstarszy w kraju wykonawca własnych wielodostępnych systemów pracujących w sieci NetWare firmy Novell.

JUNISOFTEX – to autoryzowany reseller amerykańskiej firmy Novell.

JUNISOFTEX – to dostawca zintegrowanych systemów komputerowych działających w kilkudziesięciu firmach obejmujących w każdym przedsiębiorstwie, niezależnie od formy własności i dziedziny gospodarki: finanse, majątek obrotowy, majątek trwały i nietrwały, sprzedaż, techniczne przygotowanie produkcji oraz kadry-płace.

JUNISOFTEX – to dostawca i wykonawca autoryzowanych sieci lokalnych firmy Novell.

JUNISOFTEX – to dostawca sprawdzonego sprzętu komputerowego renomowanych firm ALR, IBM, TEAM, DIGILAB, EPSON, MANNESMANN TALLY.

JUNISOFTEX – to wyłączny i autoryzowany dystrybutor komputerów i terminali firmy DIGILAB na Śląsku.

JUNISOFTEX – to dostawca i wykonawca okablowania lokalnych sieci komputerowych ARCNET, ETHERNET na kablach zwykłych i światłowodowych oraz na dwużyłowych kablach telefonicznych w przypadku stosowania oprogramowania NetWare ACCESS SERVER firmy Novell.

JUNISOFTEX – to nauczyciel, który chętnie podzieli się swoją wiedzą na organizowanych kursach w swojej szkole informatycznej – w tym nauczy Cię: podstaw informatyki, obsługi sprzętu komputerowego, eksploatacji własnych systemów informatycznych, edytorów tekstu: CHiWRITER, WordPerfect, arkuszy kalkulacyjnych QPRO v. 3.0, LOTUS 1-2-3 itp.

JUNISOFTEX – to doradca w zakresie komputerowych metod organizacji i eksploatacji systemów komputerowych.

JUNISOFTEX – to strażnik postępu i nowoczesności w Twojej firmie.

JUNISOFTEX – to gwarancja niezawodności, rzetelności i terminowości.

JUNISOFTEX – to Twój doradca i partner, któremu możesz zaufać, który Cię nigdy nie zawiedzie.

JUNISOFTEX – to partner, który Cię wysłucha i zawsze pomoże podjąć dobrą decyzję.

Jeżeli chcesz

- ★ zastosować najnowocześniejsze systemy komputerowe oraz lokalne sieci komputerowe,
- ★ zreorganizować, usprawnić i szybko skomputeryzować swoje przedsiębiorstwo,
- ★ lepiej wykorzystać już istniejące w Twoim przedsiębiorstwie komputery,
- ★ wymienić swoje niefortunnie zakupione oprogramowanie,
- ★ pozbyć się problemów eksploatacyjnych, upadających systemów, nie działających komputerów,
- ★ przeszkolić załogę swojego przedsiębiorstwa,
- ★ dobrze zainwestować swoje pieniądze i podnieść rangę swojego przedsiębiorstwa,
- ★ w przyszłości mieć dostęp do krajowej i światowej sieci komputerowej,

TO ZGŁOŚ SIĘ DO NAS!

0/17/91

Zabawa w słówka czyli kilka informacji o językoznawstwie statystycznym

Przedmiotem badań językoznawstwa są duże zbiory elementów tekstu, na przykład fonemów, wyrazów, zdań. Większość reguł językowych formułuje się na podstawie badań struktury takich zbiorów. Reguły te ujawniają prawidłowości o charakterze statystycznym. Jednym z najbardziej do tej pory rozwiniętych działów językoznawstwa statystycznego jest statystyka leksykalna, zajmująca się badaniem struktury statystycznej słownictwa.

Na podstawie zliczania różnych wyrazów w tekstach buduje się słowniki frekwencyjne, podające informacje o częstości bezwzględnej (ile razy wyraz pojawił się w badanych tekstach) i względnej (jakie jest prawdopodobieństwo pojawienia się danego wyrazu w tekście). Analiza słowników frekwencyjnych prowadzi do pewnych, dość oczywistych, wniosków. Okazuje się na przykład, że najkrótsze są wyrazy najczęściej używane.

Innym interesującym wnioskiem z analizy słowników frekwencyjnych jest zależność między liczbą wyrazów używanych w ogóle, a zasobem wyrazów tworzących większą część badanych tekstów (tab. 1). Francuski językoznawca P. Guiraud formułuje tę zależność w sposób następujący: Wyraz częsty jest najbardziej użyteczny w ogólnej sumie użyć, a rzadki – w każdym poszczególnym wypadku. Ale mówienie odbywa się zawsze w określonych okolicznościach i sytuacji, które sprawiają, że nieodzowna jest znajomość wyrazów specjalnych, odnoszących się do tej sytuacji. Wynika stąd, że samo tylko słownictwo podstawowe, jakkolwiek pokrywa całkowicie większą część dowolnego tekstu, nie pozwala zrozumieć żadnego tekstu.

Tabela 1. Zależność między częstością wyrazów, budową tekstu a ilością informacji

Liczba wyrazów na liście frekwencyjnej	Stanowi % tekstu	Zawiera % informacji
100 pierwszych	60	30
1000 pierwszych	85	50
4000 pierwszych	97,5	70
40-50 000 pozostałych	2,5	30

Statystyczne miary jakości tekstu

Niektóre statystyczne miary, opisujące własności tekstu, mają związek z jakością tego tekstu. Oczywiście nie można na podstawie wartości tych miar wnioskować o wartości tekstu w sensie jego stylu i zawartości informacyjnej. Można jednak porównywać wartości określonych współczynników dla różnych tekstów tej samej klasy i na podstawie ich porównania

z subiektywną oceną czytelności tekstu, bogactwa stylu i innych cech kwalifikujących tekst, oceniać zależność między wartością tych współczynników i jakości tekstu.

Częstość powtórzeń i współczynnik hapaksów

Każdy tekst składa się z określonej liczby jednostek, nazwijmy je wyrazami. Równocześnie niektóre wyrazy w tekście powtarzają się – będziemy je nazywać słowoformami. Zdanie: *mój dom to mój pałac* zawiera pięć wyrazów, ale tylko cztery słowoformy.

Oznaczmy przez W liczbę wyrazów, natomiast przez S liczbę słowoform występujących w tekście. Nazywając z grecka hapaksami (od: *hapax legomenon* – występujący pojedynczo) wyrazy, które pojawiają się w całym tekście tylko raz, oraz określając ich liczbę jako h , możemy wyliczyć następujące współczynniki, mające związek z czytelnością tekstu:

średnia częstość powtórzeń słowoform = W/S ,
oraz współczynnik hapaksów = h/S

Dla wyrazów, które powtarzają się w tekście k razy podamy jeszcze jeden współczynnik:

częstość względną wyrazu = k/W

Jak się okazuje, najistotniejsze znaczenie dla oceny jakości tekstów ma współczynnik hapaksów. Stwierdzono korelację między wartością tego współczynnika a czytelnością tekstów.

Współczynnik Zipfa

Jeśli uporządkujemy wyrazy występujące w tekście według liczby ich wystąpień (inaczej częstości bezwzględnej) f_1 , to uzyskamy listę, na której pozycję wyrazu nazwiemy jego rangą r_1 . Już w 1916 r. francuski stenograf J. Estoup zauważył, że dla danych całej listy iloczyn $f_1 \star r_1$ jest wielkością mniej więcej stałą. Dalsze badania, prowadzone przez amerykańskiego językoznawcę G. K. Zipfa, dowiodły istnienia empirycznej zależności, wyrażającej się wzorem:

$$f_1 = C \star r_1^{-r}$$

Po podzieleniu obydwu stron przez liczbę wyrazów w tekście N oraz obustronnym zlogarytmizowaniu, otrzymujemy funkcje:

$$\log p_r = k - \Gamma \star \log r$$

gdzie:

p_r – empiryczne prawdopodobieństwo wystąpienia wyrazu o randze r w tekście,

r – ranga,

k – wartość stała, charakterystyczna dla tekstu,

Γ – współczynnik Zipfa.

Wartość Γ nazywana współczynnikiem Zipfa jest miarą bogactwa słownika tekstu. Wartość bezwzględna tego współczynnika jest tym większa, im bardziej zróżnicowane jest słownictwo tekstu.

Entropia i informacja

Jeśli X jest zmienną losową dyskretną o rozkładzie danym przez ciąg:

$$p_1 = P\{X = X_1\}$$

to wartość:

$$H(X) = \sum p_1 \log_2 p_1$$

jest nazywana entropią zmiennej losowej X . Jeśli teraz przyjmiemy, że prawdopodobieństwo p_1 wystąpienia wyrazu X_1 w tekście praktycznie jest równe jego częstości względnej, to fizyczna interpretacja wartości $H(X)$, jest taka, że teoretycznie istnieje możliwość zbudowania takiego kodu binarnego, w którym dla zapisania dowolnego elementu zbioru wyrazów w badanym tekście, potrzeba będzie H bitów. Innymi słowy, entropia zmiennej X jest proporcjonalna do bogactwa stylistycznego i informacyjnego badanego tekstu. Tekst składający się z jednego wyrazu ma entropię równą zero. Entropia może być, obok współczynnika Zipfa, traktowana jako miara bogactwa stylu lub zawartości informacji w tekście.

Dyspersja wyrazów w tekście

Kolejną statystyczną miarą, opartą o wyrazy traktowane jako ciągi znaków, jest dyspersja wyrazów w tekście. Można założyć, że w dobrze skonstruowanym tekście, autor pisząc o jednym aspekcie sprawy używa określonego fragmentu swojego słownika, natomiast gdy przechodzi do omawiania innego aspektu problemu jego słownik nieco się zmienia. Ta tendencja może być zmierzona przez obserwację wyrazów powtarzających się w tekście dokładnie dwa razy. Jeśli istnieje tendencja do zmiany słownika, to średnia odległość takich wyrazów – którą to wartość nazwiemy dyspersją wyrazów w tekście – jest mniejsza niż by wynikało to z losowego rozkładu wyrazów w tekście.

Indeksy czytelności

W końcu lat czterdziestych R. Flesch opublikował w *Journal of Applied Psychology* wyniki badań nad zależnością między łatwością przyswojenia tekstu pisanego, a jego strukturą statystyczną. Inni językoznawcy amerykańscy opracowali w latach siedemdziesiątych kilka wzorów podających zależność między średnią długością wyrazów i zdań w tekście, a czytelnością tego tekstu. Jakkolwiek formuły te mają sens jedynie dla języka angielskiego, to warto przytoczyć wzory, na podstawie których oblicza się wartości tych *indeksów czytelności* tekstu:

Formuła Kincaida:

$$IC = 11.8 \star S.. + .39 \star W - 15.59$$

ARI (Automated Readability Index):

$$IC = 4.71 \star L.. + 5 \star W_z - 21.43$$

Formuła Colemana-Liau:

$$IC = 5.89 \star L_w + 3 \star Z_{100^w} - 15.8$$

Formuła Flescha:

$$IC = 206.835 - 84.6 \star S_w - 1.015 \star W_z$$

gdzie:

IC – indeks czytelności,

L_w – średnia liczba liter przypadających na wyraz,

S_w – średnia liczba sylab przypadających na wyraz,

W_z – średnia liczba wyrazów przypadających na zdanie,

Z_{100^w} – średnia liczba zdań przypadających na 100 wyrazów.

Badania z pogranicza językoznawstwa statystycznego i psychologii wykazały faktyczną zależność między łatwością przyswojenia i zrozumienia tekstu, a powyższymi indeksami.

Statystyczna analiza poprawności wyrazów

Większość współczesnych programów przygotowania tekstu ma opcję, umożliwiającą kontrolę ortograficznej poprawności używanych wyrazów. W tekstach przygotowywanych przy użyciu komputera, oprócz błędów ortograficznych występuje na ogół wiele tzw. literówek, spowodowanych opuszczeniem litery, dodaniem litery lub naciśnięciem niewłaściwego klawisza. Najczęstszą metodą kontroli tekstu pod tym kątem jest stosowanie słownika, zawierającego kilka do kilkudziesięciu tysięcy wyrazów, i sygnalizowanie wyrazów nie występujących w słowniku. Główny koszt stosowania tej metody wynika z konieczności opracowania słownika. Równocześnie słownik uniwersalny jest mało przydatny dla tekstów specjalistycznych, zawierających wyrazy nie używane w języku potocznym.

Do wykrywania wyrazów prawdopodobnie błędnych można zastosować inną metodę, opartą na statystyce powtórzeń w tekście grup liter. Najprostszą metodą jest zliczanie wszystkich grup sąsiednich dwu i trzech liter. Nazwiemy takie grupy odpowiednio digramami i trigramami. Wyraz *niespełnienie* zawiera następujące trigramy i digramy:

1 ★ # ni	1 ★ # n
3 ★ nie	3 ★ ni
1 ★ ies	3 ★ ie
1 ★ esp	1 ★ es
1 ★ spe	1 ★ sp
1 ★ pel.	1 ★ pe
1 ★ eln	1 ★ el
1 ★ lni	1 ★ ln
1 ★ ien	1 ★ en
1 ★ eni	1 ★ e #
1 ★ ie #	

Znak # w powyższym zestawieniu oznacza dowolny separator. Jak widać, w statystyce uwzględniamy również początkowe i końcowe litery wyrazu. Badany wyraz liczy 13 liter. Można z niego wyodrębnić 13 trigramów (w tym 11 różnych) oraz 14 digramów (10 różnych). Statystyczny rozkład digramów i trigramów jest związany ze słownictwem używanym w tekście. Istnieje kilka algorytmów wykrywania wyrazów potencjalnie błędnych, przy użyciu digramów i trigramów.

Najprostszy z tych algorytmów oparty jest na spostrzeżeniu, że błąd literowy (a takie najczęściej poprawia się na komputerach) powoduje zaburzenia w trzech kolejnych trigramach. Według tego podejścia potencjalnie błędne są wyrazy zawierające obok siebie dwa albo trzy trigramy występujące w badanym tekście rzadko. Miarą rzadkości może być – na przykład – średnia statystyczna powtórzeń trigramów w tekście.

Bardziej złożony jest algorytm wykorzystujący tzw. współczynnik dziwności wyrazu. Jeśli mamy trigram *xyz* o częstości

wystąpień w tekście równej $f(xyz)$, złożony z digramów xy i yz , o częstościach odpowiednio $f(xy)$ i $f(yz)$, to indeks dziwności tego trigramu wynosi:

$$[\log(f(xy) - 1) + \log(f(yz) - 1)]/2 - \log(f(xyz) - 1)$$

Dla celów obliczeniowych przyjmuje się wartość $\log(0)$ równą -10 . Współczynnik dziwności wyrazu jest średnią kwadratową indeksów trigramów wyrazu.

Zaletą metody opartej na statystyce digramów i trigramów jest to, że program wykorzystujący tę technikę może działać nie tylko na dokumentach wykorzystujących różne style i zakresy słownictwa fachowego, ale wręcz jest niezależny od języka. Równie dobre efekty daje jego działanie dla tekstów zarówno w języku polskim jak i w angielskim.

Połączenie powyższej techniki z zastosowaniem słownika, umożliwiającego odfiltrowanie z listy wyrazów podejrzanych

Tabela 2

Liczba wystąpień kolejnych liter alfabetu	Stanowi % tekstu	Częstość występowania znaków	Najczęstsze digramy, trigramy i wyrazy	
A*854	A - 8.57	ST*227	54*ch	59*w
A*123	I - 7.19	W*225	52*po	36*wyrzów
B*100	E - 7.01	A*214	47*nie	35*i
C*427	O - 6.31	IE*183	42*ie	30*tekstu
C*45	W - 5.96	E*175	42*y ch	28*tekście
D*259	T - 5.74	NI*173	34*ści	26*jest
E*698	N - 5.49	W*161	31*Pr	22*się
E*122	Z - 5.37	T*153	29*st	21*z
F*60	Y - 5.05	RA*148	29*sta	19*na
G*141	R - 4.69	S*140	28*ej	14*dla
H*102	S - 4.66	P*135	28*ość	14*o
I*716	C - 4.29	TE*133	27*ego	12*wyrzów
J*248	K - 3.57	I*129	25*wy	11*to
K*356	D - 2.60	CZ*124	25*czn	10*słownika
L*214	P - 2.51	Y*120	25*go	10*współczynnik
Ł*126	J - 2.49	ZY*112	24*ia	9*a
M*217	M - 2.18	ÓW*108	24*jąc	9*że
N*547	L - 2.15	O*105	24*owa	8*do
N*23	U - 1.91	NA*104	24*y st	8*jego
O*629	O - 1.63	WY*103	23*na	8*Liczba
O*162	G - 1.42	Z*102	23*Za	8*N
P*250	Ś - 1.33	CI*101	22*nia	8*średnia
Q*0	Ł - 1.26	EK*101	22*ów	8*tekstów
R*467	A - 1.23	OW*96	22*sty	8*trigramów
S*464	Ę - 1.22	WA*93	21*in	8*wyrzów
S*133	H - 1.02	J*91	21*ają	7*digramów
T*572	B - 1.00	YC*90	21*ani	7*między
U*190	Z - 0.68	CH*89	20*eni	7*tęgo
V*1	F - 0.60	AZ*87	20*tyc	7*wyrz
W*594	C - 0.45	TY*85	20*wan	7*X
X*18	N - 0.23	ŚC*84	19*my	7*znaków
Y*503	X - 0.18	I*83	18*Prz	6*f
Z*535	V - 0.01	H*82	18*wie	6*log
Ż*68	Q - 0.00	N*79	18*zna	6*tych
Ź*3	Z - 0.00	PO*79	17*ni	6*wartości
		OŚ*74	17*dzi	6*wartość
		A*73	17*nik	6*zależność
		U*73	17*nyc	6*Zipfa
		O*72	17*wni	5*częstości
		JE*72	17*ycz	5*h
		D*70	16*te	5*IC
		AN*68	16*awi	5*informacji
		IA*66	16*gra	5*Jeśli
		TO*66	16*łów	5*językoznawstwa
		KS*65	16*na	5*k
		YR*65	16*ym	5*L
		TA*64	15*sł	5*można
		ZA*64	15*ący	5*nie
		PR*61	15*ci	5*podstawie
		WI*61	15*em	5*powtórzeń

wyrazy poprawne, pozwala na szybkie i dość skuteczne wstępne poprawienie tekstów znacznej objętości w bardzo krótkim czasie.

Przykład statystyk

Poniżej zostanie zaprezentowany przykład niektórych statystyk i rozkładów na podstawie tekstu tego właśnie artykułu. Przyjmijmy, że tekst ten zawiera 1591 wyrazów, składających się z 9964 znaków, co odpowiada 6.42 stronom znormalizowanego maszynopisu. Współczynnik Zipfa dla tekstu wynosi 0.89, a współczynnik hapaksów 0.72. Niski poziom wartości tych współczynników wynika z niewielkiej objętości tekstu i jego tematycznej jednorodności.

Tabela 3

Długość wyrazu	Liczba w tekście	Wystąpień w słowniku	Średnia powtórzeń
1	191	19	10.05
2	117	39	3.00
3	104	39	2.67
4	106	44	2.41
5	136	82	1.66
6	186	97	1.92
7	193	87	2.22
8	142	85	1.67
9	130	88	1.48
10	93	74	1.26
11	65	53	1.23
12	49	29	1.69
13	28	24	1.17
14	33	22	1.50
15	7	7	1.00
16	4	4	1.00
17	4	4	1.00
18	3	1	3.00
Razem znaków	9964	6009	
Razem wyrazów	1591	798	1.99
Znaków/wyraz	6.26	7.53	

W tabeli 2 przedstawiono statystyki znaków, digramów, trigramów i wyrazów w tekście. Druga kolumna informuje, jaki procent tekstu stanowi użycie danego znaku. Pozostałe podają informacje o częstości wszystkich znaków i pięćdziesięciu najczęstszych digramów, trigramów i wyrazów.

Tabela 4

Liczba wyrazów	Powtarza się razy	Stanowi %	
		tekstu	słownika
1	59	3.71	0.13
1	36	5.97	0.25
1	35	8.17	0.38
1	30	10.06	0.50
1	28	11.82	0.63
1	26	13.45	0.75
1	22	14.83	0.88
1	21	16.15	1.00
1	19	17.35	1.13
2	14	19.11	1.38
1	12	19.86	1.50
1	11	20.55	1.63
2	10	21.81	1.88
2	9	22.94	2.13
8	8	26.96	3.13
6	7	29.60	3.88
7	6	32.24	4.76
18	5	37.90	7.02
16	4	41.92	9.02
47	3	50.79	14.91
104	2	63.86	27.94
575	1	100.00	100.00

Rozkład długości wyrazów w tekście ilustruje tabela 3. Słownikiem nazywamy listę wszystkich wyrazów, występujących w tekście. Długość wyrazu to liczba jego znaków. Statystykę pokrycia tekstu ilustruje tab. 4.

Jak widać, nawet w tak niewielkim tekście jest spełniona reguła nierównomiernego pokrycia tekstu i słownika. Wyrazy stanowiące 1% słownika, pokrywają ok. 16% tekstu, podczas gdy 15% słownika wystarczy do pokrycia połowy tekstu.

LITERATURA

- [1] Bojar B.: Zarys językoznawstwa dla informatyków. Wydawnictwa UW. Warszawa 1976.
- [2] Bolc L., Cichy M., Różańska L.: Przetwarzanie języka naturalnego. WNT, Warszawa 1982
- [3] Guiraud P.: Zagadnienia i metody statystyki językoznawczej. PWN, Warszawa 1966
- [4] McMahon L. E., Cherry L. L., Morris R.: Statistical Text Processing, The Bell System Technical Journal, Vol. 57, No. 6, July-August 1978
- [5] Miodunka W.: Podstawy leksykologii i leksykografii. PWN Warszawa 1989
- [6] Sambor J.: Językoznawstwo statystyczne dla pracowników informacji naukowej. Materiały szkoleniowe. CINTiE, Warszawa 1978
- [7] Sambor J.: Słowa i liczby. Zagadnienia językoznawstwa statystycznego. Ossolineum, Wrocław 1972.

Wszyscy Autorzy, których teksty znajdują się w Redakcji, a jeszcze nie zostały opublikowane proszeni są o przesyłanie do nas informacji, o których szczegółowo piszemy na stronie 29.

 <p>SPÓŁKA z o.o.</p>	 <p>UNITED MICROELECTRONICS CORPORATION</p>	 <p>HEWLETT PACKARD COMPONENTS</p>	
<ul style="list-style-type: none"> • CZĘŚCI ELEKTRONICZNE • KOMPUTERY PS/1, PS/2 • DRUKARKI HP • INSTALACJE SIECI KOMPUTEROWYCH 	<ul style="list-style-type: none"> • UKŁADY PAMIĘCI • UKŁADY KOMPUTEROWE • UKŁADY KOMUNIKACYJNE I KOMERCYJNE 	<ul style="list-style-type: none"> • TRANSOPTORY • WSKAŹNIKI ŚWIETELNE • WYŚWIETLACZE LED • PRODUKTY KODÓW KRĘKOWYCH • KONTROLERY I CZUJNIKI RUCHU • TECHNIKA ŚWIATŁOWODOWA • ELEMENTY W CZ. I MIKROFALOWE • PODZESPOŁY DO MONTAŻU POCZERCIKOWEGO (SMI) 	<ul style="list-style-type: none"> • POTENCJOMETRY TRIMPOT • HYBRIDY REZYSTOROWE • REZYSTORY SUBMINIATUROWE • BEZPIECZNIKI MULTIFUSE • POTENCJOMETRY PRECYZYJNE • POTENCJOMETRY PANELE CZOŁOWYCH I KODERY • CEWKI I TRANSFORMATORY • CZUJNIKI CIŚNIENIA, POŁOŻENIA I PRZYŚPIESZENIA
Partnerzy handlowi: ANALOG DEVICES, IIT, MOTOROLA, SAMSUNG, TELEFUNKEN i inni	PRZEDSTAWICIELSTWO	DYSTRYBUCJA	DYSTRYBUCJA
			
 <p>sp. z o.o.</p>			
<p>00-194 Warszawa, ul. Długa 4 tel. (02) 6352263, 6352264 fax (02) 6352195, ttx 816075</p>			

SO 575/91

Nowe techniki w konstruowaniu interfejsów użytkownika

dokończenie ze s. 14

Przy zastosowaniu technik programowania obiektowego i zdefiniowaniu standardowych interfejsów (kanałów komunikacji między obiektami) jest możliwe zbudowanie środowiska w którym tworzenie programów użytkowych odbywałoby się w ogóle bez programowania (w tradycyjnym tego słowa znaczeniu) a polegałoby na interakcyjnym montowaniu standardowych modułów w agregaty realizujące konkretne funkcje użytkowe. Technika ta (ang. *scripting*) jest pewną analogią do bardzo elastycznego rozwiązania przyjętego w procesorach poleceń systemu UNIX (*sh* i *csh*), pozwalającego na budowanie poleceń dokonujących złożonego przetwarzania za pomocą standardowych programów UNIX-a łączonych mechanizmami komunikacji międzyprocesowej.

Rozwiązaniem, które może dawać pewien obraz tego, jak w przyszłości będzie wyglądało konstruowanie oprogramowania, jest Interface Builder, będący częścią standardowego oprogramowania dostarczanego w każdej z serii roboczych NeXT. Program użytkowy jest w nim budowany przez wybór zestawu bibliotecznych obiektów interakcyjnych i określenie pewnych relacji funkcjonalnych między nimi, co powoduje powiązanie zbioru obiektów w jeden funkcjonujący automat.

Bardzo obiecujące są też rozwiązania typu HyperCard, będące skrzyżowaniem arkusza kalkulacyjnego z technikami obiektowymi. Programowanie polega tu na stopniowym specyfikowaniu zachowań oraz interfejsów do stosów tzw. kart, czyli obiektów mających własności przetwarzania informacji.

W dziedzinie bliższej fizycznej realizacji interfejsów przyszłość może należeć do inteligentnych serwerów programowanych w specjalizowanych językach typu Display PostScript i ich następcach z wbudowanymi mechanizmami realizującymi wyższe funkcje logiczne interfejsu użytkownika. Przykładem interesującego rozwiązania jest tu pakiet HyperNews [2] opracowany w Instytucie Turinga w Glasgow.

* * *

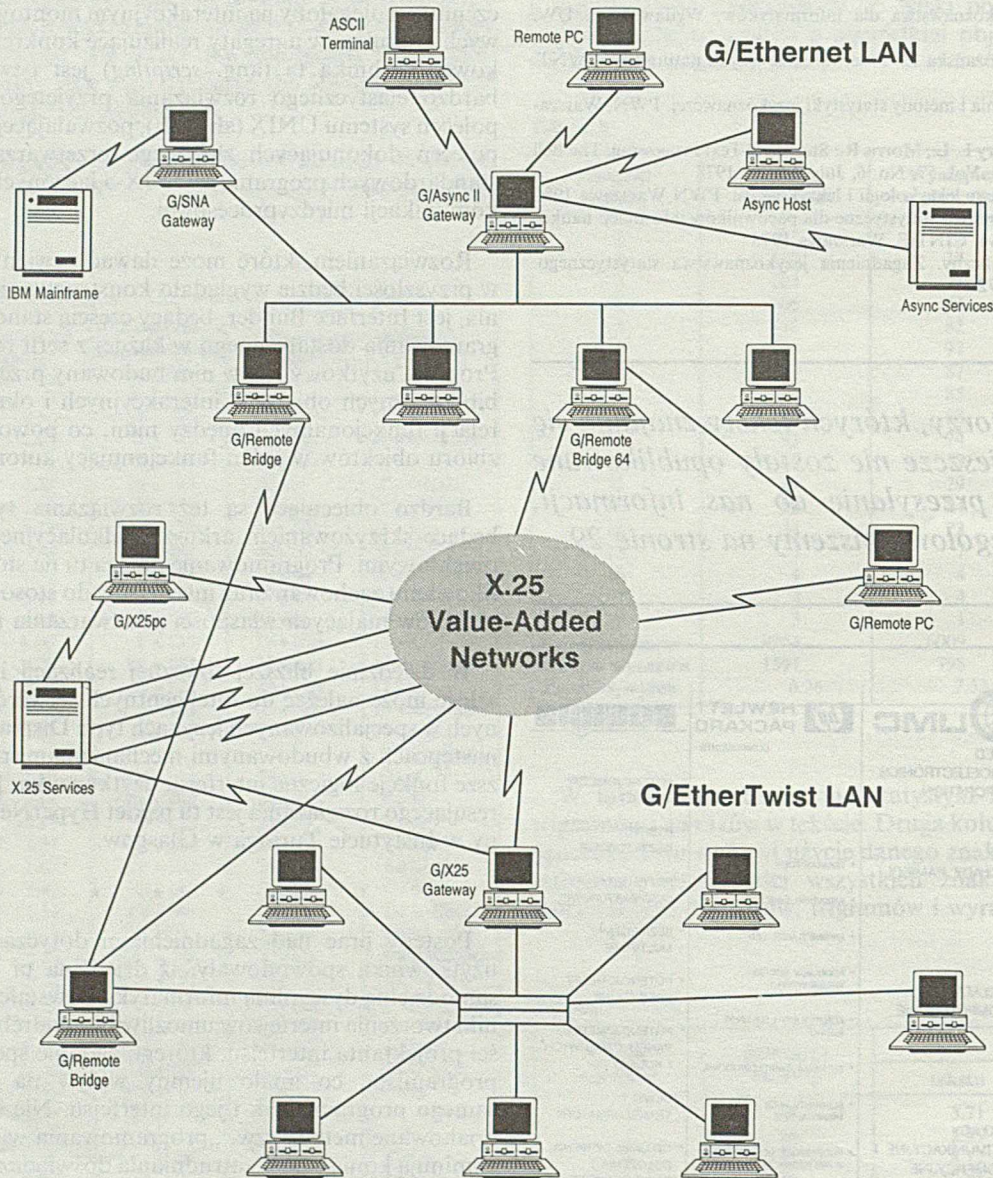
Postępy prac nad zagadnieniami dotyczącymi interfejsów użytkownika spowodowały, iż dziedzina ta stała się niejako samodzielną dyscypliną informatyki. Powstałe narzędzia i techniki tworzenia interfejsów umożliwiły wyodrębnienie specjalności projektanta interfejsu, którego zadanie spełniał dotychczas programista, co miało ujemny wpływ na jakość zarówno samego programu, jak i jego interfejsu. Nie są jeszcze jednak opanowane metody tzw. „programowania wizualnego”, które eliminują konieczność zatrudniania doświadczonych programistów do tworzenia programów użytkowych. Znane rozwiązania nie są jeszcze na etapie umożliwiającym ustanowienie jakiegokolwiek standardu, takiego jak Motif lub Open Look w swojej klasie. W rzeczywistości prace nad takim oprogramowaniem są dopiero w fazie eksperymentów.

LITERATURA

- [1] Mikes S.: Two Goopy Builders That Stick. UNIX WORLD, January 1991
- [2] Pearce D.: HyperNews: An Interactive Interface Design Tool. IEEE Colloquium on User Interface Management Systems, 1989
- [3] PostScript Language Reference Manual. Adobe Systems, Inc.
- [4] Xlib - C language X interface, Version 11. Massachusetts Institute of Technology
- [5] X Window System Protocol, Version 11. Massachusetts Institute of Technology.

Produkty sieciowe LAN i WAN

Gateway
communications, inc.



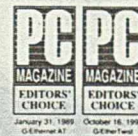
BEZKONKURENCYJNE PRODUKTY
SIECI KOMPUTEROWYCH NAGRADZANE PRZEZ:
PC MAGAZINE, LAN MAGAZINE, INFO WORLD
OFERUJE AUTORYZOWANY DYSTRYBUTOR:



MIKROB

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



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

0/15/91

Strukturalna analiza i projektowanie systemów informatycznych – metodyka Gane'a-Sarsona (2)

W drugiej części artykułu omówiono trzeci etap modelowania logicznego [6] – podejścia do tworzenia systemów informatycznych opracowanego przez Gane'a i Sarsona.

Modelowanie danych w procesie tworzenia systemu informatycznego

W dziedzinie modelowania danych istnieje wiele różnych poziomów opisu bazy danych. Na przykład w ramach ANSI/X3/SPARC wyróżnia się trzy poziomy opisu bazy danych: fizyczny (wewnętrzny), pojęciowy oraz zewnętrzny [20]. Jeśli patrzeć na proces modelowania danych z punktu widzenia użytkownika (podejście infologiczne [8,18]), to model danych służy do dokumentowania potrzeb informacyjnych (wymagań) użytkownika i pomaga w lepszym zrozumieniu przez ludzi modelowego wycinka rzeczywistości (dziedziny przedmiotowej). Powinien on ujmować całą semantykę danych.

Pierwszym etapem w procesie projektowania schematu bazy danych jest analiza wymagań informacyjnych użytkowników, w ramach której identyfikuje się funkcje dziedziny przedmiotowej (istotne rodzaje działalności, procesy), oraz klasy danych wymagane lub tworzone przez te funkcje. Wymagania użytkowników zostają następnie wyrażone jako opis przedsięwzięcia za pośrednictwem odpowiedniego modelu danych. Następne dwa etapy po akceptacji opisu przedsięwzięcia (tj. opisu bazy danych i projektowanie fizycznej bazy danych) są związane z jego przeniesieniem na opis bazy danych zgodny z modelem danych wbudowanym w docelowy DBMS.

Definicja i podstawowe pojęcia dotyczące modelu danych

Głównym celem modelowania danych jest uzyskanie za pomocą pojęć struktury, więzów i operacji możliwie wiernego opisu rzeczywistości (dziedziny przedmiotowej).

Model danych składa się ze zbioru reguł generujących G oraz zbioru operacji O [18]. Reguły generujące G można podzielić na dwie grupy:

- generatory struktury GS (specyfikacja struktury),
- generatory GC – więzy konieczne ze względu na semantykę i integralność.

Generatory struktury

Specyfikacji struktury można dokonać np. za pomocą pojęcia atrybutu, typów encji i typów związków, dla których podstawą są zbiory i relacje.

Zbiory charakteryzują się dwoma podstawowymi własnościami: definiującą (intensja zbioru) oraz reprezentującą (ekstensja zbioru), która określa aktualne wystąpienie zbioru. W modelowaniu danych używa się abstrakcji w celu nadania interpretacji zbiorom obiektów [8, 17, 18].

Relacja R jest uważana za typ obiektu definiujący zbiór krotek złożonych z konkretnych (tj. określonych wystąpień obiektów). Relacji R można przypisać różne interpretacje, np. przypisując każdej krotce relacji R określoną encję.

Encję (tj. wystąpienie encji) można określić jako jednoznacznie identyfikowalny składnik badanej rzeczywistości [22], coś co istnieje lub może być pomyślane i jest odróżnialne.

Stosując do encji abstrakcję tworzy się ogólny typ encji (kategorię), odpowiadający kategoriom obiektów świata rzeczywistego. Typ encji (intensja relacji) odpowiada agregacji atrybutów. Zbiór encji (ekstensja relacji) odpowiada wszystkim aktualnym wartościom związanych z każdym z atrybutów typu encji.

Relację R można interpretować jako związek między oddzielnymi typami encji. Intensją takiej relacji jest typ związku (czyli agregacja typów encji), ekstensją zaś zbiór związków. Typ związku może być typem związku dwuargumentowego – łączącego jedynie dwa typy encji lub n -argumentowego.

Więzy

Specyfikacje struktury GS ujmują tylko część własności danych. Inne własności określa się specyfikując więzy GC , tj. nakładając dodatkowe ograniczenia na wartości danych i sposób ich wzajemnego powiązania. Wyróżnia się dwa typy więzów:

- **więzy wbudowane**, będące integralną częścią struktur danych (np. zakaz powielania krotek w relacjach),
- **więzy jawne** (tzw. semantyczne więzy integralności, asercje – np. klucze).

Więzy powinny być zachowane w każdej strukturze modelu danych, z którą są związane.

Bardzo ważną własnością w modelowaniu danych, przydatną do definiowania więzów jest tzw. **moc odwzorowania**. Każda dwuargumentowa relacja R definiuje dwa odwzorowania:

$$R: S1 \rightarrow S2 \text{ oraz } R': S2 \rightarrow S1.$$

Dla każdego odwzorowania określa się minimalną i maksymalną moc, odpowiadającą liczbie obiektów z $S1$, które mogą być

powiązane z obiektami z $S2$ i odwrotnie. Przez nałożenie konkretnych ograniczeń na minimalną i maksymalną moc odwzorowania otrzymujemy różne typy odwzorowań (całkowite, częściowe, jednoznaczne 1:1, funkcyjne 1:N, wieloznaczne $M:N$). Są to tzw. więzy przynależności [18]. Dotyczą one zazwyczaj typów związków i określają wymagania co do odwzorowania, które musi zostać zdefiniowane między encjami, aby jedna z nich lub obie mogły istnieć.

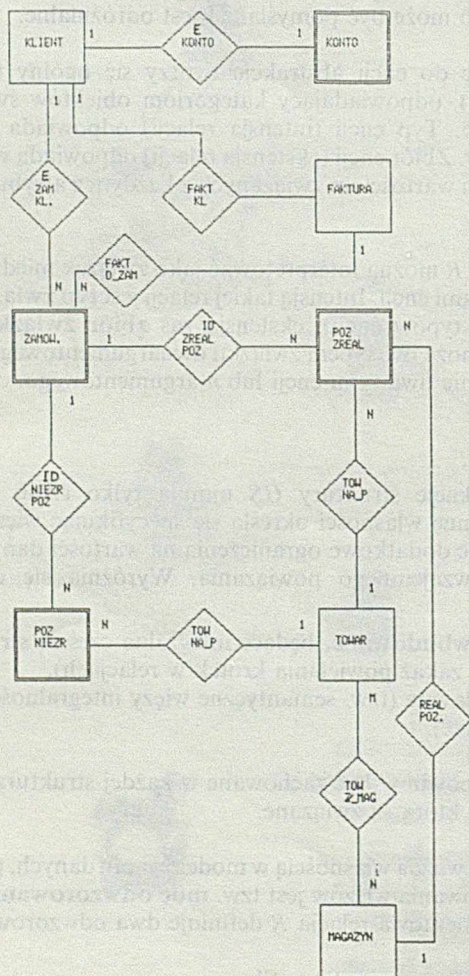
Operacje

Operacje 0 modelu danych (zwane językiem danych) definiowane są jako funkcje częściowe określone na stanach bazy danych. Odwzorowują one jeden stan bazy danych w inny lub niezdefiniowany (jeśli zostają naruszone więzy jawne).

Modele danych związków encji

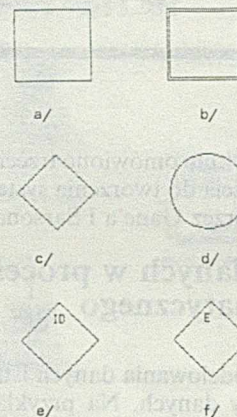
Są one przeznaczone przede wszystkim dla potrzeb analizy i projektowania systemów. Są to modele „wyższego poziomu” w porównaniu z modelami relacyjnym, sieciowym lub hierarchicznym, gdyż umożliwiają specyfikację więzów jawnych bezpośrednio w modelu danych oraz reprezentowanie typów związków wieloznacznych.

Najbardziej popularnym z nich jest model danych związków encji Chena (model ER – ang. *entity-relationship*) [1, 2] stanowiący dokumentację logicznych własności bazy danych.



Rys. 1. Diagram związków encji dla bazy danych systemu dystrybucji towarów

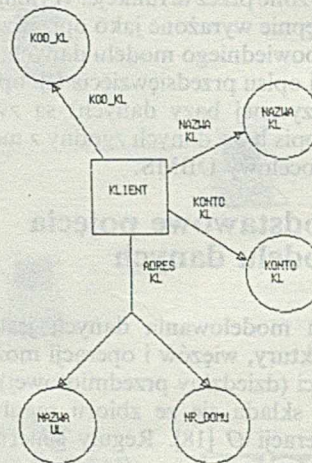
Strukturami podstawowymi w tym modelu są: typ encji oraz typ związku, reprezentujący ogólną strukturę powiązań między zbiorami encji. Strukturę bazy danych zorganizowaną zgodnie z modelem związków encji przedstawia się za pomocą diagramu związków encji (ERD – ang. *entity-relationship diagram*). Na rysunku 1 przedstawiono diagram związków encji dla bazy z uproszczonego systemu dystrybucji towarów (opisywanego w pierwszej części artykułu), symbolikę diagramu przedstawiono natomiast na rys. 2 (notacja Chena). W modelu ER powiązania mogą reprezentować n -argumentowy związek między zbiorami encji i mogą być jednoznaczne, funkcyjne i wieloznaczne. Dozwolone są powiązania rekurencyjne.



Rys. 2. Symbolika diagramu związków encji – notacja Chena:

- a) zbiór encji regularnej,
- b) zbiór encji słabej,
- c) zbiór związku regularnego,
- d) zbiór wartości (atrybutu),
- e), f) przykłady związku słabego ID -zależność i więzy istnienia E

Maksymalną dozwoloną moc zbioru encji w zbiorze związku podaje się w sposób jawny na krawędzi łączącej zbiór związku z zbiorem encji (rys. 1).



Rys. 3. Atrybuty i zbiory wartości dla zbioru encji *KLIENT*

Atrybut jest odwzorowaniem między zbiorem encji (lub zbiorem związku) a zbiorem wartości i nadaje pewną interpretację zbiorowi. Przykład odwzorowań atrybutowych dla zbioru encji *KLIENT* z diagramu ER przedstawionego na rys. 1 pokazano na rys. 3.

Jeśli występowanie encji zależy od występowania innej encji, to taki zależny zbiór encji nazywa się **zbiorem encji słabej**. Zbiór związku obejmujący zbiór encji słabej nazywa się **zbiorem związku słabego**. Większość więzów w modelu ER to więzy jawne.

W przypadku atrybutów można definiować zbiór wartości atrybutu (rys. 4 – więzy W2). Można również nakładać ograniczenia na wartości atrybutu występujące w bazie danych (rys. 4 – więzy W3). W celu jednoznacznego identyfikowania encji w zbiorze encji można specyfikować klucz encji (rys. 4 – więzy W1). Zbiory związku mają również swoje klucze, złożone z kluczy zbiorów encji składowych.

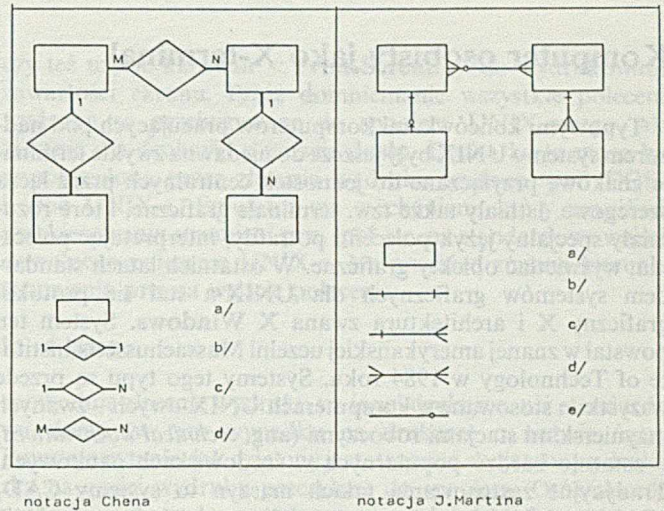
WIĘZY

- W1 : klucze : kod_klienta kod_towaru konto_kl nr_magaz
- W2 : $1 < nr_magaz < 4$
- W3 : wartość ZAMOW. równa się sumie wartości POZ ZREAL
- W4 : każdemu klientowi odpowiada tylko jedno konto
- W5 : każda encja typu ZAMOW. jest powiązana z jedną encją typu KLIENT (więzy istnienia E)
- W6 : każdej encji typu POZ ZREAL musi odpowiadać jedna encja typu ZAMOW. (ID-zależność)

Rys. 4. Przykładowe więzy bazy danych z rys. 1

W modelu ER można jawnie reprezentować więzy istnienia oraz ID-zależność. Więzy istnienia oznaczają, że występowanie encji zależy od występowania związanej z nią encji, tj. usunięcie encji pociąga za sobą usunięcie związanej z nią encji słabej (rys. 4 – więzy W5). Jeśli natomiast do jednoznacznego identyfikowania encji w zbiorze encji jest konieczny związek z innymi

encjami, to takie więzy nazywa się ID-zależnością (rys. 4 – więzy W6). Stanowią one automatyczne więzy istnienia.



Rys. 5. Porównanie notacji Chena z notacją J. Martina:

- a) typ encji
- b) powiązanie 1:1,
- c) powiązanie 1:N,
- d) powiązanie M:N,
- e) opcjonalność zbioru encji w zbiorze związku

Istnieje kilka różnych notacji diagramów związków encji – na rysunku 5 pokazano porównanie notacji Chena z notacją J. Martina zwanej czasami notacją „wronich łapek” (ang. „crow's foot” notation).

/unix + dos

Integracja systemów operacyjnych MS-DOS i UNIX (3)

Jeżeli użytkownik zdecyduje się na maszynę UNIX-ową jako następcę sieci PC, zainstalowanie na nowym minikomputerze serwera sieci jest bardzo wygodnym i eleganckim rozwiązaniem:

- zachowane zostaje dotychczasowe środowisko pracy użytkowników;
- nie ma potrzeby instalowania dodatkowej sieci terminali znakowych czy też dodatkowego okablowania – istniejące połączenia umożliwią dostęp do nowego systemu z każdego komputera osobistego przyłączonego do sieci;
- możliwe jest natychmiastowe wykorzystanie części przestrzeni dyskowej UNIX-a jako rozszerzenia pojemności dysków sieciowych;
- możliwa jest jednoczesna eksploatacja starych aplikacji i wdrażanie nowego systemu – na tym samym sprzęcie;
- użytkownicy sieci PC oprócz nowego serwera zyskują możliwość skorzystania z możliwości UNIX-a – szczególnie jeśli chodzi o jego właściwości komunikacyjne;
- nowa aplikacja może być rozproszona: np. wyszukiwanie

w bazie danych odbywa się na serwerze UNIX-owym, a wprowadzanie i walidacja oraz prezentacja wyników jest wykonywana na komputerach osobistych; jest to możliwe dzięki specjalnym interfejsom programowym zawartym w pakietach sieciowych systemów operacyjnych dla UNIX-a, umożliwiającym przezroczysty dostęp do danych i komunikację między procesami MS-DOS-a i UNIX-a

Należy zdawać sobie sprawę, że sprawność dedykowanego serwera sieciowego systemu operacyjnego będzie zawsze większa, niż serwera UNIX-owego. Ten ostatni, oprócz udostępniania swoich zasobów otoczeniu (co jest praktycznie jedynym zajęciem dedykowanego serwera) wykonuje normalne zadania maszyny UNIX-owej. W związku z tym nie należy uważać, że serwery UNIX-owe mogą w pełni zastąpić klasyczne serwery dedykowane. Tam, gdzie najlepszym rozwiązaniem jest sieć lokalna PC, dedykowany serwer będzie zawsze najkorzystniejszy, zapewniając szybki dostęp do dysków sieciowych.

Komputer osobisty jako X-terminal

Typowymi końcówkami komputerów pracujących pod nadzorem systemu UNIX były jeszcze do niedawna zwykłe terminale znakowe przyłączane do jednostek centralnych przez łącza szeregowo. Istniały także tzw. terminale graficzne, które rozumiały specjalny język poleceń i potrafiły, interpretując polecenia, wyświetlać obiekty graficzne. W ostatnich latach standardem systemów graficznych dla UNIX-a stał się protokół graficzny X i architektura zwana X Windows. System ten powstał w znanej amerykańskiej uczelni Massachusetts Institute of Technology w 1984 roku. Systemy tego typu są przede wszystkim stosowane w komputerach UNIX-owych – zwanych inżynierskimi stacjami roboczymi (ang. *technical workstations*) – ostatnio bardzo popularnych w środowiskach naukowych. Tradycyjne zastosowania takich maszyn to systemy CAD/CAM i najróżniejsze systemy specjalizowane, które prezentują wyniki swego przetwarzania na kolorowym ekranie o dużej rozdzielczości (co najmniej 1000 × 1000 punktów).

System X Windows jest implementacją modelu „klient/serwer”. Jego składnikami są:

X Serwer – urządzenie wyposażone w monitor, klawiaturę, myszkę oraz możliwość komunikacji z procesami-klientami; serwer na zlecenie klienta (przekazywane za pomocą specjalnego protokołu o nazwie X) tworzy i wyświetla na ekranie takie obiekty, jak okienka i ikony, natomiast w okienkach może wyświetlać znaki i obrazy graficzne (zawsze przekazywane w postaci mapy bitowej). X Serwer przekazuje klientom (także przy użyciu protokołu X) informację o naciskanych przez użytkownika klawiszach, o ruchach myszki oraz „klikaniu” przyciskanych klawiszy.

X Klient – aplikacja, czyli proces wykonywany w systemie operacyjnym, który komunikuje się z X Serwerem i zleca mu wyświetlanie wyników obliczeń. X Klient komunikuje się z użytkownikiem za pomocą X Serwera; każdy serwer może obsługiwać jednocześnie wiele procesów klientów, które mogą wykonywać się na różnych komputerach.

X Serwer może być zaimplementowany na tym samym komputerze, na którym wykonują się procesy X Klientów. Jest to typowa sytuacja w przypadku UNIX-owych stacji roboczych. Może to być także oddzielne urządzenie przyłączone razem z komputerami X Klientów szybką siecią lokalną. X Klient, zrealizowany jako oddzielne urządzenie, nosi nazwę X Terminala. Sprzęt ten jest oferowany na rynku przez wiele firm.

W typowej konfiguracji stacje graficzne są zawsze połączone ze sobą szybką siecią lokalną – można do niej dołączyć maszyny UNIX-owe, nie mające wbudowanych X Serwerów, oraz X Terminale. Takie środowisko jest zupełnie przezroczyste dla aplikacji – proces klienta pracujący na dowolnym komputerze może korzystać z dowolnych X Serwerów znajdujących się w sieci (zarówno samodzielnych, jak i wbudowanych w poszczególne komputery). Najczęściej wykorzystywanym standardem sieci lokalnej jest Ethernet i rodzina protokołów TCP/IP – protokół X jest przenoszony w sieci przez protokół datagramowy UDP. Bardzo rzadko stosuje się także łącza szeregowo i implementację tej samej rodziny protokołów dla tego medium – SLIP, czyli *Serial Line Internet Protocol*. Jednak prędkość transmisji łącza

szeregowego jest zwykle za mała dla potrzeb przesyłania map bitowych obrazów o dużej rozdzielczości.

Analogicznie do emulowania terminala znakowego, komputer osobisty może także pełnić funkcję X Terminala. Aby sprawność emulacji była zadowalająca PC powinien mieć sporą moc obliczeniową (co najmniej procesor 80286 16 MHz lub 80386/80486) oraz rozszerzoną pamięć operacyjną (co najmniej 1 MB) i dysk stały. Oprócz tego niezbędne jest następujące wyposażenie dodatkowe:

- adapter graficzny o dużej rozdzielczości – najlepiej VGA lub SVGA; w ostateczności może być adapter EGA;
- odpowiedni do używanej karty graficznej monitor – kolorowy lub monochromatyczny;
- trójprzyciskowa myszka;
- adapter sieciowy Ethernet.

Oprogramowanie służące do emulatora X Terminala na PC jest dostarczane na rynek przez wielu producentów. Zwykle pakiety te wymagają uprzedniego zainstalowania na komputerze osobistym modułu TCP/IP, niezbędnego do nawiązania połączenia z maszyną UNIX-ową. Oczywiście, aby dołączenie X Terminala miało sens, w sieci – do której będzie on przyłączony – musi istnieć co najmniej jeden komputer UNIX-owy – węzeł sieci TCP/IP – z zainstalowanym pakietem umożliwiającym pracę X Klientów.

Emulacja środowiska MS-DOS w UNIX-ie

Systemy operacyjne MS-DOS jest tak szeroko używanym środowiskiem, że emulowanie go w innych systemach jest powszechne wśród producentów komputerowych. Teoretycznie jest to możliwe w dowolnej architekturze sprzętu, natomiast szczególnie łatwo jest to osiągnąć na maszynach wyposażonych w procesory Intel 80386/80486. Procesory te (oprócz tego, że są zgodne z procesorami 8086/8088/80286) mają możliwość pracy w tzw. wirtualnym trybie 8086: proces pracujący w takim trybie ma do dyspozycji wszystkie rejestry i instrukcje procesora 8086 (wykonywane bezpośrednio) oraz jego przestrzeń adresową, która jest faktycznie odwzorowywana na pamięć wirtualną rzeczywistego procesora 80386/486. Natomiast w przypadku implementacji UNIX-a na innych procesorach, dla emulacji MS-DOS-a jest niezbędne emulowanie procesora 8086 przez interpretację jego instrukcji maszynowych i ich zamianę na ciągi instrukcji maszynowych procesora macierzystego.

Emulacja MS-DOS-a polega na emulacji środowiska sprzętowego komputera typu PC w systemie UNIX przez odpowiednią obsługę wywołań systemowych DOS (INT 21H) i BIOS – ich przekazanie do wykonania po odpowiedniej konwersji macierzystemu systemowi operacyjnemu. Na przykład, wszystkie wywołania systemowe dotyczące pamięci dyskowej i systemu plików MS-DOS-a (emulowanego przez system plików UNIX-a) muszą być przekazane do wykonania UNIX-owi, ponieważ fizyczne operacje dyskowe odbywają się na systemie plików UNIX-a. Ale – na przykład – wywołania dotyczące dyskietki w formacie DOS-owskim są wykonywane bezpośrednio przez kod systemu MS-DOS emulatora.

Dokładniejsze omówienia mechanizmów i zasad emulacji zostanie dokonane przez skupienie się na poszczególnych aspektach pracy emulatora. Zostanie to zrobione na przykła-

dzie emulatora o nazwie Merge firmy Locus Computing Corp. Bardzo podobne właściwości funkcjonalne ma także emulator o nazwie VP/ix wchodzący w skład systemu Open Desktop – implementacji UNIX-a firmy Santa Cruz Operations.

Pamięć dyskowa

Emulator potrafi korzystać z trzech rodzajów pamięci dyskowej:

- 1) systemu plików UNIX-a;
- 2) systemu plików MS-DOS-a na dyskietkach i partycjach DOS-owskich na dyskach twardych;
- 3) wirtualnych partycji i dyskietek DOS-owskich symulowanych przez pliki UNIX-a.

Po uruchomieniu emulatora system plików UNIX-a jest odwzorowywany na DOS-owski dysk C.: Tak więc np. plik znany UNIX-owi jako:

```
/usr/local/bin/kermit
```

będzie dostępny dla emulatora MS-DOS-a jako:

```
C:/USR/LOCAL/BIN/KERMIT
```

Jedną z niezgodności pomiędzy MS-DOS-em a UNIX-em, znaną już z wcześniejszej części niniejszego opracowania dotyczącej serwerów plików, jest inny format nazwy pliku w obu systemach. UNIX dopuszcza w nazwie do 14 dowolnych znaków (kropka jest takim samym składnikiem nazwy, jak każdy inny znak) i znaczące są dla niego małe i duże litery (ang. *case sensitivity*). Natomiast MS-DOS ich nie rozróżnia, a nazwa pliku składa się z części głównej (o długości do 8 znaków) i rozszerzenia (o długości do 3 znaków). Przestrzeń nazw DOS-owskich jest podzbiorem nazw standardu UNIX-a. W związku z tym emulator musi sobie radzić z nazwami zbiorów UNIX-owych niezgodnych ze standardem DOS-owskim – są one poddawane konwersji polegającej na arbitralnej zamianie kilku znaków i odrzuceniu nadmiarowej (z punktu widzenia DOS-a) części nazwy. Np. plik UNIX-a o nazwie:

```
MiXcAsE.xtnded
```

będzie dostępny poprzez emulator MS-DOS-a jako:

```
MIXCA*5U.XTN
```

Kłopotów z konwersją nazw można w większości przypadków uniknąć, nadając plikom podczas pracy pod UNIX-em nazwy spełniające wymagania MS-DOS-a. Niestety niektóre narzędzia UNIX-owe (np. *lex* i *yacc*) generują nazwy niezrozumiałe dla DOS-a.

Dostęp do UNIX-owego systemu plików z emulatora jest regulowany w taki sam sposób, jak przy bezpośredniej pracy z UNIX-em – przestrzegane są UNIX-owe prawa dostępu do plików dla poszczególnych użytkowników korzystających ze środowiska MS-DOS.

Drukarki

Emulator umożliwia korzystanie ze wszystkich DOS-owskich poleceń dotyczących drukowania, takich jak **PRINT**, **COPY**

czy też użycie klawisza < **Print-Screnn** > do wydrukowania zawartości ekranu. Przez domniemanie wszystkie polecenia wydruku są przekazywane do spoolera UNIX-owego jako zlecenia dla drukarki o nazwie **doslp**. Drukarka ta może być współdzielona przez zlecenia wydruku wydane spod DOS-a i spod UNIX-a. Emulator ma mechanizm, który umożliwia przekierowania strumieni **prn**, **lpt1** etc. na standardowe wejście dowolnego polecenia UNIX-a. Na przykład zmiana kierunku strumienia **prn** na wejście polecenia:

```
lp -d laser
```

spowoduje zlecenie wydruku na innej drukarce, niż standardowa **doslp** (w tym wypadku na drukarce o nazwie *laser*). Powstaje pytanie, jak dzielony jest strumień skojarzony z drukarką na części wydruku przesyłane do spoolera. Otóż zlecenie wydruku jest wysyłane, gdy zostanie spełniony jeden z następujących warunków:

1. Działanie programu, który generował wydruk zostało zakończone.
2. Uplęło co najmniej 15 sekund od chwili, gdy program wysłał ostatni znak na drukarkę.

Taka zasada zapobiega sytuacji generowania wielu zleceń wydruku (wraz z nagłówkami), gdy program DOS-owski wysła na drukarkę wiele porcji znaków w krótkich odstępach. Jeśli odstępy te są krótsze niż 15 sekund, to do spoolera zostanie wysłane tylko jedno zlecenie wydruku. Wartość parametru definiującego opóźnienie (ang. *timeout*) można zmienić.

Istnieją sytuacje, gdy współpraca ze spoolerem UNIX-a jest niepożądana. Emulator daje wtedy możliwość bezpośredniej obsługi portu drukarki przez system operacyjny MS-DOS. Drukarka, której port będzie obsługiwany bezpośrednio przez emulator, powinna zostać zdezaktywowana w podsystemie drukującym UNIX-a za pomocą polecenia **disable**. Umożliwia to uniknięcie przemieszczania wydruków w przypadku, gdy będzie tej drukarki używał jednocześnie program DOS-owski i spooler UNIX-a.

PAWEŁ ŻAK

W związku z wprowadzeniem powszechnego podatku dochodowego i koniecznością informowania właściwych Urzędów Skarbowych o wszystkich wypłatach zwracamy się do Autorów wysyłających teksty do opublikowania o podawanie następujących danych:

- nazwisko, imiona (pierwsze i drugie),
- imiona: ojca i matki,
- miejsce i data urodzenia,
- numer identyfikacyjny PESEL (wpisany przez Biuro Meldunkowe do dowodu osobistego – nie mylić z numerem dowodu osobistego!),
- dokładny adres (miejsce zameldowania),
- adres Urzędu Skarbowego właściwego dla miejsca zamieszkania Autora (bardzo ważne – bez tej informacji nie możemy przekazać honorarium do wypłaty!).

Prosimy także – do naszej wiadomości – podać numer telefonu służbowego i ustalić formę przekazania honorarium (kasa Wydawnictwa, poczta, konto).

Perspektywa systemów 64-bitowych

Pierwsze komputery osobiste wykorzystywały mikroprocesory o 8-bitowych rejestrach liczb całkowitych – stąd nazwano je systemami 8-bitowymi. Potem przyszła era systemów 16-bitowych, następnie pojawiły się systemy 32-bitowe, a ostatnio – perspektywa 64-bitowych. Określenia te nie są w pełni jednoznaczne, gdyż do tych rejestrów są doprowadzane i odprowadzane za pomocą magistrali danych, a adresy – określające położenie danych – za pomocną magistrali adresowych. Magistrale te mogą mieć inne szerokości (liczbę bitów) niż rejestry. Np. IBM PC z procesorem 8088 miał 16-bitowe rejestry, ale 8-bitową magistralę danych i 20-bitową magistralę adresową. Podobnie układ 386SX tym różni się od w pełni 32-bitowego 386DX, że podobnie jak Motorola 68000, ma 16-bitową magistralę danych i 24-bitową magistralę adresową. Zamieszczona poniżej tabela 1 zawiera 20 wybranych realizacji jednostek centralnych (procesorów) z okresu ostatnich dwudziestu kilku lat.

Większość współczesnych komputerów wykorzystuje dziś procesory 32-bitowe. Następną generacją – procesory 64-bitowe pozwolą osiągnąć jeszcze większe moce obliczeniowe w konstrukcjach biurowych. Obecnie tylko Mips R4000 jest mikroprocesorem przekraczającym zakres 32-bitowy, zarówno w danych, jak i w adresowaniu, lecz jest to niewątpliwie trend przyszłościowy. Przewiduje się, że mikroprocesory te będą dominować na rynku już w 1995 roku, a być może – nawet wcześniej. Są one bowiem atrakcyjne zarówno ze względu na przetwarzanie słów 64-bitowych, jak i możliwość wykorzystania 32-bitowego obszaru adresowego. Zastosowania dla takich mikroprocesorów istnieją zarówno wśród procesorów usługowych (serwerów) jak i autonomicznych systemów biurowych (ang. *desktop*).

Zamieszczona tabela 1 rozróżnia liczbę bitów danych i adresowych w ramach architektury listy rozkazów, co reprezentuje punkt widzenia programisty pracującego w języku symbolicznym oraz sprzętowej realizacji tej architektury. Kolejne standardy architektury nie zmieniały się lub rozwijały zgodnie z istniejącymi tendencjami. Realizacje natomiast kierowa-

ły się względami ekonomicznymi. Niektóre systemy stosują pewne formy segmentacji dla uzyskania większej przestrzeni adresowej (adres wirtualny). Natomiast liczba bitów magistrali danych może być różna dla tego samego typu jednostki centralnej. Wielkość rejestru zmienno-przecinkowego, zawierającego prawie zawsze 64 lub 80 bitów, nie jest tu podawana.

W niektórych zastosowaniach użycie dłuższego rejestru liczb całkowitych znacznie przyspiesza obliczenia. Dotyczy to zwłaszcza przetwarzania długich ciągów liczb oraz niektórych zastosowań graficznych i arytmetycznych, takich jak szyfrowanie czy obliczenia finansowe.

Jeszcze bardziej istotne jest zwiększenie obszaru adresowego ponad 32 bity. Dotyczy to wielkich baz danych wykorzystujących dyski o pojemności kilku Gbajtów. Dla większości komputerów ograniczenia pamięci wirtualnej przekraczają ograniczenia fizyczne, ponieważ najprościej jest powiększyć pojemność istniejącej pamięci. Ale gdy ograniczenia wirtualne są mniejsze od fizycznych, oprogramowanie nie może wykorzystać zwiększenia pojemności pamięci.

W epoce drogich pamięci rdzeniowych wystarczały 24 bity adresowe, co ograniczało pojemność pamięci wirtualnej (i fizycznej) do 16 Mbajtów. W owym czasie nawet ówczesne „duże” systemy, takie jak IBM 360/75 miały pamięć o pojemności 1 Mbajta, a największe jak 360/91 – 6 MB. Gdy pamięci stały się tańsze, a w systemie S/370 wprowadzono adresowanie wirtualne, wystąpiła potrzeba rozszerzenia przestrzeni adresowej i dlatego w 1983 r. mikroprocesory 370XA miały już adresy 31-bitowe, co jednak w połowie lat osiemdziesiątych dla niektórych zastosowań, zwłaszcza baz danych, stało się również niewystarczające.

Wśród minikomputerów ograniczeniem były przydzielane dla poszczególnych zadań obszary danych i rozkazów o pojemności 64 kB, mimo iż niektóre z egzemplarzy komputera PDP-11/70 miały pamięć operacyjną o pojemności 4 MB. Ograniczenie to znikło przy systemie VAX-11/780, gdzie nie było już potrzeby dzielenia dużych programów na części.

Mikrokomputery, nawet 16-bitowe, miały mechanizm segmentacji umożliwiający przydział większych obszarów

Tabela 1

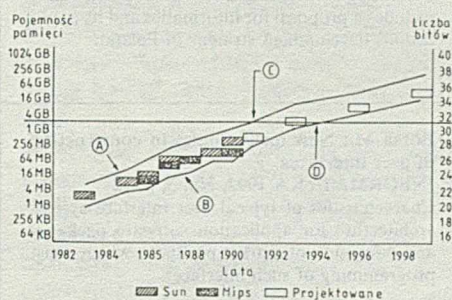
Jednostka centralna (procesor)	Rok	Liczba bitów systemu	Architektura listy rozkazów		Realizacja sprzętowa	
			Długość rejestru	Długość adresu	Długość adresu	Magistrala danych
IBM S/360	1964	32	32	32	24	8-128
DEC PDP-11/45	1973	16	16	16	18	32
DEC PDP-11/70	1976	16	16	16	22	32
DEC VAX-11/780	1978	32	32	31	32	64
Motorola 6800	1980	32	32	24	24	16
IBM S/370XA	1983	32	32	31	32	128
Intel 286DX	1985	32	32	32	32	32
Motorola 68020	1985	32	32	32	32	32
HP Precision	1986	32	32	32	32	32-64
Mips R2000	1986	32	32	31	32	32
Intel 386SX	1987	32	32	32	24	16
Motorola 68030	1987	32	32	32	32	32
Sun SPARC	1987	32	32	32	36	32-64
IBM ESA/370	1988	32	32	31	32	128
Intel 860	1989	64	32	32	32	64
Intel 486DX	1989	32	32	32	32	32
IBM RISC System/6000	1990	32	32	32	32	64-128
Mips R4000	1990	64	64	40-62	36	64
Motorola 68040	1990	32	32	32	32	32
Intel 486SX	1991	32	32	32	32	32

adresowych. Programiści komputerów osobistych umieli posługiwać się tymi możliwościami. Mikroprocesor Motorola MC6800 miał wprawdzie 32-bitowy rejestr liczb całkowitych, ale do adresowania wykorzystywał tylko 24 bity, co z czasem stało się istotnym ograniczeniem. Pozostałe osiem bitów było wykorzystywane przez programistów do różnych celów, a kiedy mikroprocesor MC68020 wykorzystywał do adresowania wszystkie 32 bity – programy takie nie mogły być na nim realizowane. Podobne problemy wystąpiły przy przejściu z pierwotnego McIntosha na model Mac II.

Nie ulega wątpliwości, że pojemność pamięci systemów komputerowych ciągle rośnie. Próbuje się określić matematyczne reguły tego wzrostu. Przybliżone zależności wykazują, że pojemność kostek pamięciowych DRAM (dynamicznych) wzrasta czterokrotnie co 3 lata, a wykorzystanie pamięci wirtualnej wzrasta o współczynnik 1,5–2 każdego roku. Najłatwiej i najtaniej jest dołączyć pamięć dodatkową, ale tylko wówczas, gdy oprogramowanie może to w prosty sposób wykorzystać. Gdy naturalna wielkość kodu i danych osiągnie, a następnie przekroczy, ograniczenie adresu wirtualnego, wzrasta stopień trudności programowania, ponieważ wtedy trzeba tworzyć struktury sztuczne. Jest to szczególnie niewygodne w sytuacji, gdy ograniczenie adresu wirtualnego jest znacznie mniejsze od ograniczenia fizycznego, gdyż nawet powiększenie pamięci nie poprawia istniejącej sytuacji. Na szczęście ograniczenie adresu wirtualnego jest na ogół większe od ograniczenia fizycznego i wtedy, przy zbyt małej wydajności, dobre wyniki daje powiększenie pamięci.

Nie ma ustalonej wartości stosunku ograniczeń adresu wirtualnego dla maksymalnego zadania do ograniczeń fizycznych. Przyjmuje się, że zadowolająca jest tu wartość 4, o ile akceptuje to system operacyjny. Niektórzy programiści uważają, że jest to zbyt mało i dlatego nowoczesne techniki zarządzania zbiorami (jak Multics lub Mach) osiągają ten parametr na poziomie zbliżonym do 8. Np. przy projektowaniu i symulacji mikromodułów elektronicznych, serwery o pojemności pamięci 256 MB obsługują programy wykorzystujące obrazy wirtualne o pojemności 2 GB. Obecnie są oferowane systemy biurkowe o pojemności 128 MB, a po wprowadzeniu kostek o pojemności 16 Mb osiągną one 512 MB, co pozwoli realizować programy wykorzystujące 4 GB pamięci wirtualnej.

Rysunek pokazuje historyczny rozwój systemów firm Sun i Mips Computer oraz systemów projektowanych przez te firmy na najbliższe lata. Z rysunku tego wyraźnie widać, że już obecnie ujawniają się ograniczenia 32-bitowej przestrzeni adresowej, dotyczące jednak tylko systemów technologicznie najbardziej zaawansowanych. W 1993 lub 1994 r. ograniczenia te będą odnosić się również do pamięci fizycznych. Wkrótce bowiem, dzięki dostępności kostek 16-Mbitowych, będą sprzedawane systemy z pamięcią operacyjną 2–4 GB co będzie można łatwo uzyskać przez wymianę kart pamięci w istniejących obudowach. Jeśli nie uczynią tego projektanci wymienionych firm, gotowi są to uczynić projektanci z takich firm, jak Silicon Graphics, Hewlett-Packard czy IBM. Odpowiednie układy jednostek centralnych muszą być gotowe już w 1991 r., aby w 1992 r. uruchomić produkcję odpowiedniego sprzętu, a w latach 1983–1994, – zastosowania.



Historyczny rozwój systemów firm Sun i Mips Computer oraz systemów projektowanych

A – największy pojedynczy typowy program czterokrotnie przekracza pojemność maksymalnej pamięci fizycznej

B – wiele systemów dostarcza się z połową maksymalnej pamięci fizycznej

C – kłopoty z architekturą 32-bitową dla czołowych systemów komputerowych

D – kłopoty z architekturą 32-bitową dla wielu systemów komputerowych (1994)

Przejście z 32-bitowych rejestrów liczb całkowitych bezpośrednio na rejestry 64-bitowe (a nie wykorzystywanie wielokrotności pośrednich, np. rejestrów 48-bitowych) jest podyktowane m.in. szerokim rozpowszechnieniem języka C, który poprawnie działa na maszynach adresowanych bajtowo, gdzie liczba bajtów 8-bitowych jest potęgą 2. Zastępowanie 64 bitów segmentacją nie daje również zadowalających wyników.

Pośród zastosowań, które najwcześniej będą wymagać powiększenia obszaru adresowego pamięci, należy wymienić:

Bazy danych. Współczesne systemy operacyjne stosują przekształcenia, w których całe zbiory są bezpośrednio odwzorowywane w pamięci wirtualnej zadania. Ponieważ można pozostawić wolną przestrzeń pamięciową przewidując wzrost rozmiarów zbiorów, pamięć wirtualna jest znacznie szybciej zużywana niż fizyczna. Ze wzrostem wydajności jednostek centralnych w porównaniu do czasu dostępu na dysku, unika się tego dostępu przez utrzymywanie plików dyskowych w dużych, szybkich pamięciach notatnikowych (ang. *cache*). Administratorzy baz danych realizowanych na dużych komputerach zaczęli odczuwać to wtedy, gdy instalacje przekroczyły granicę 240 bajtów. Projekty systemów rozproszonych często wykorzystując określone bity adresu jako adres węzła, a pozostałe – jako adres w węźle;

Obrazy. Przy 300 elementach na cal i 24-bitowym kolorze, strona o rozmiarze $8,5 \times 11$ cali ($215,9 \times 279,4$ mm) wykorzystuje 25 MB pamięci, a więc 4 GB pamięci pozwala zapamiętać jedynie 160 takich stron. Bazy danych takich obiektów szybko stają się bardzo duże;

Wideo. Obraz wizyjny bez kompresji o 24-bitowym kolorze i ekranie 1280×1024 elementy potrzebuje 3,75 MB pamięci. Przy 24 klatkach na sekundę, 4 GB pamięci wystarczą jedynie na 45 sekund programu;

CAD. Systemy projektowania wspomaganego komputerem często obejmują duże sieci z wieloma stanowiskami i serwerami, które zarządzają bazami danych i dokonują wielkich symulacji. Oprogramowanie 64-bitowe jest tu oczywiście pożądane, a chociaż w mniejszych systemach nie występują tak wielkie ilości danych, to ze względu na zgodność oprogramowania obejmuje ono całe sieci;

Systemy informacji geograficznych. Systemy te są kombinacją map, obrazów i innych danych. Mają podobne właściwości jak systemy wizyjne i CAD;

Duże obliczenia numeryczne. Dla tych zastosowań każde zwiększenie pojemności pamięci jest bardzo pożądane. Projektanci takich systemów zawsze natrafiają na ograniczenia tej pojemności.

W tabeli 2 pokazano różne zastosowania oraz przyczyny, dla których użycie systemów 64-bitowych przynosi znaczne korzyści. Wyodrębniono także serwery i małe systemy biurkowe.

Wprowadzenie systemów 64-bitowych pozwala realizować nowe, duże zastosowania również na małych konfiguracjach sprzętowych. Gdy systemy te staną się szerzej dostępne, wiele programów bę-

dzie wymagało stopniowego dostosowania. I chociaż nie ma dla nich tak wielu obszarów zastosowań, to są jednak systemy bardzo użyteczne, np. przy projektowaniu dużych układów elektronicznych.

Tabela 2

Zastosowanie	Serwery		Małe systemy	
	Szybkość	Obszar adresowy	Szybkość	Zgodność
Operacje bajtowe	x		x	
Grafika			x	
Duże liczby całkowite	x		x	
Bazy danych		x		x
Video			x	
Obrazy		x		x
CAD		x		x
Informacje geograficzne		x		x
Duże obliczenia numeryczne		x	x	

Wraz z powszechnym wprowadzeniem układów 64-bitowych ich cenne własności, związane z szybką arytmetyką działań na dużych liczbach oraz z największym obszarem pamięci wirtualnej, staną się bardziej oczywiste. Obecnie istnieją już systemy operacyjne i programy użytkowe, pozwalające wykorzystać te własności.

JAN RYŻKO

<p>Praca zbiorowa: Propozycja strategii rozwoju informatyki i jej zastosowań w Rzeczypospolitej Polskiej (3) INFORMATYKA 1992, nr 3, s. 1 Trzecia część oraz podsumowanie całości opracowania Polskiego Towarzystwa Informatycznego dla Urzędu Rady Ministrów, zawierającego propozycję strategii rozwoju informatyki i jej zastosowań w Polsce.</p>	<p>Collective work: Proposal for informatics and its application development strategy in Poland (3) INFORMATYKA 1992, No. 3, p. 1 Third part and summary of the Polish Informatics Society's elaboration for the Cabinet, which includes a proposal for informatics and its applications development strategy in Poland.</p>	<p>Gemeinschaftsarbeit: Ein Vorschlag für die Entwicklung der Informatik und ihrer Anwendungen in Polen (3) INFORMATYKA 1992, Nr. 3, S. 1 Dritter Teil und Zusammenfassung von einer Bearbeitung der Polnischen Informatik Gesellschaft für das Ministerrat, die ein Vorschlag für die Entwicklung der Informatik und ihrer Anwendungen in Polen umfasst.</p>
<p>Borsa M.: Nowe techniki w konstruowaniu interfejsów użytkownika INFORMATYKA 1992, nr 3, s. 12 Charakterystyka typowej architektury systemu interfejsu użytkownika dla pakietów oprogramowania użytkowego oraz omówienie dostępnych na rynku produktów wspomagających programowanie takich interfejsów.</p>	<p>Borsa M.: New technologies in constructing of user interfaces INFORMATYKA 1992, No. 3, p. 12 Characteristics of typical user interface system architecture for application software packages and discussion of market products, which assist programming of such interfaces.</p>	<p>Borsa M.: Neue Techniken in Konstruieren von Anwenderschnittstellen INFORMATYKA 1992, Nr. 3, S. 12 Eine Charakteristik von typischer Architektur des Anwenderschnittstellesystems für Pakette der Anwendungssoftware und eine Besprechung von Marktprodukten, die Programmierung solcher Schnittstellen unterstützen.</p>
<p>Baranowski J.: Od BETRIEVE-a do SQL-a. Nowoczesne, wielopoziomowe środowisko baz danych firmy NOVELL INFORMATYKA 1992, nr 3, s. 15 Charakterystyka środowiska baz danych firmy NOVELL, zawierająca omówienie podstawowych cech produktów tworzących to środowisko, a także wniosków praktycznego ich wykorzystywania.</p>	<p>Baranowski J.: From BTRIEVE to SQL. Modern multilevel data base environment of NOVELL INFORMATYKA 1992, No. 3, p. 15 Characteristics of the NOVELL data bases environment, which includes discussion of basic features for products creating this environment, as well as of conclusions for practical application of the products.</p>	<p>Baranowski J.: Von BTRIEVE bis SQL. Moderne mehrschichtige Datenbankumwelt von NOVELL INFORMATYKA 1992, Nr. 3, S. 15 Eine Charakteristik des NOVELL-Datenbankumweltas, die eine Besprechung der Grundeigenschaften von umweltgestaltenden Produkten, sowie Schlussfolgerungen ihrer paktischen Ausnutzung, umfasst.</p>
<p>Fuglewicz P.: Zabawa w słówka czyli kilka informacji o językoznawstwie statystycznym INFORMATYKA 1992, nr 3, s. 20 Charakterystyka metod komputerowego badania struktury statystycznej słownictwa na przykładzie eksperymentalnej analizy struktury słownictwa użytego w tekście autora artykułu.</p>	<p>Fuglewicz P.: Playing with words or some informations on statistic linguistics INFORMATYKA 1992, No. 3, p. 20 Characteristics of the vocabulary's statistic structure computer assisted investigation on example of experimental analysis of vocabulary used in the author's paper.</p>	<p>Fuglewicz P.: Wörtleinspiel oder einige Informationen über statistische Linguistik INFORMATYKA 1992, Nr. 3, S. 20 Eine Charakteristik von Methoden einer computerunterstützten Forschung der statistischen Struktur eines Wortschatzes auf Beispiel einer experimentalen Analyse der Wortschatzstruktur, die im Artikel des Autors verwendet wurde.</p>
<p>Stapor K.: Strukturalna analiza i projektowanie systemów informatycznych – metodyka Gane'a-Sarsona (2) INFORMATYKA 1992, nr 3, s. 25 Druga część charakterystyki metody Gane'a-Sarsona, stosowanej do strukturalnej analizy i projektowania systemów informatycznych, zawierająca omówienie trzeciego etapu modelowania logicznego.</p>	<p>Stapor K.: Structural analysis and data processing system designng – the Gane-Sarson method (2) INFORMATYKA 1992, No. 3, p. 25 Second part of characteristics of the Gane-Sarson method, which is applied for structural analysis and data processing system designing. The part contains discussion of logical modelling third stage.</p>	<p>Stapor K.: Strukturanalyse und Projektierung der EDV-Systeme – die Gane-Sarson-Methode (2) INFORMATYKA 1992, Nr. 3, S. 25 Zweiter Teil einer Charakteristik von Gane-Sarson-Methode, die zur Strukturanalyse und Projektierung der EDV-Systeme verwendet wurde. Der Teil umfasst eine Besprechung der dritten Etappe logischer Modelierung.</p>

II krajowa konferencja metodyki nauczania przedmiotów informatycznych w uczelniach rolniczych

W dniu 6 grudnia 1991 roku w Akademii Rolniczej w Szczecinie odbyła się konferencja na temat metodyki nauczania przedmiotów informatycznych w uczelniach rolniczych. Jej organizatorami byli: Międzyuczelniany Ośrodek Metodyczny Akademii Rolniczych przy SGGW-AR w Warszawie oraz Zakład Informatyki Wydziału Ekonomiki i Organizacji Gospodarki Żywnościowej Akademii Rolniczej w Szczecinie.

Pierwsza konferencja tego typu odbyła się dwa lata temu w Akademii Rolniczej (Wrocław). Od tego czasu nastąpiły istotne zmiany w systemie zarządzania gospodarką żywnościową, powodując w tej dziedzinie wzrost znaczenia informatyki, a także rozszerzenie zakresu jej zastosowań na takie obszary jak marketing, a nawet prowadzenie indywidualnego gospodarstwa rolnego. Wobec zmian modelu gospodarczego kraju również uczelnie rolnicze mogą obecnie tworzyć własne modele programowe i organizacyjne studiów.

Konferencja miała na celu znalezienie elastycznych i optymalnych rozwiązań systemowych, umożliwiających unowocześnienie procesu studiów oraz zaspokojenie prostudyjnych zainteresowań studentów w zakresie nowego systemu zarządzania. Wśród 67 jej uczestników byli nie tylko nauczyciele akademicy, ale również studenci oraz osoby decydujące o rozwoju informatyki w uczelniach.

Celami konferencji było nauczanie informatyki i przedmiotów informatycznych w poszczególnych akademiach oraz prezentacja rozwiązań zastosowań informatyki w dydaktyce i pracach badawczych. Podstawę do dyskusji i wymiany doświadczeń stanowiły następujące referaty:

W części I – kształtowanie treści programowych:

- Nauczanie informatyki w nowych warunkach ekonomicznych kraju prof. dr hab. Jadwiga Orylska (AR-Szczecin);
- Wielopoziomowe nauczanie przed-

miotów informatycznych w uczelniach rolniczych – prof. dr hab. Mirosław Dytczak (SGGW-AR Warszawa).

W części II – osiągnięcia i zastosowania:

- Zastosowanie systemów eksportowych w rolnictwie – stan bieżący, potrzeby i perspektywy – prof. dr hab. Mirosław Dytczak (SGGW-AR – Warszawa);
- Sztuczna inteligencja i systemy eksportowe w zarządzaniu przedsiębiorstwami gospodarki żywnościowej – prof. dr hab. Jadwiga Orylska (AR-Szczecin);
- Relacyjne bazy danych w technice komputerowej. Pokaz: system dydaktyczny z zastosowaniem relacyjnym baz danych – mgr Liwiusz Siemianowski (AR-Szczecin);
- Metoda ustalania i oprogramowania parametrów technologicznych produkcji roślinnej jako element systemów ekspertowych w rolnictwie (pokaz) – prof. dr hab. Ludosław Drelichowski i współpracownicy (ATR-Bydgoszcz);
- Wielokryterialne modele i metody wspomagające procesy decyzyjne w rolnictwie (pokaz) – prof. dr hab. Bogdan Krawiec (AR-Szczecin);
- Zastosowanie metody symulacji planowanej do weryfikacji algorytmów modeli genetycznych populacji na przykładzie parametrów odziedziczalności w hodowli buraka cukrowego (pokaz) – prof. dr hab. Ludosław Drelichowski i współpracownicy (ATR-Bydgoszcz);
- Zastosowanie analizy obrazów w systemach eksportowych (pokaz) – mgr Andrzej Sulawiak (AR-Szczecin);
- Wykorzystanie rozszerzonych możliwości wybranego systemu CAD do opracowania rysunków schematycznych linii aparaturowych – mgr Dariusz Piotrowski (SGGW-AR Warszawa);
- Zintegrowany pakiet ENABLE (pokaz) – dr Czesław Cegłowski (AR-Szczecin);
- System informacji gospodarczej woj. szczecińskiego (pokaz) – mgr Lesław Róź (AR-Szczecin).

Referaty oraz pokazy zawierały koncepcje rozwojowe oparte na dotychczasowych

doświadczeniach dydaktycznych i przewidywanych zmianach na rynku pracy absolwentów. Sytuacja nauczania przedmiotów informatycznych jest w wielu uczelniach trudna, ponieważ z jednej strony następuje redukcja liczby godzin przeznaczonych na dydaktykę tych przedmiotów, a z drugiej strony rośnie nacisk studentów zainteresowanych tym właśnie kierunkiem wiedzy.

Dyskusja pozwoliła sformułować następujące wnioski:

- zmiany w systemie ekonomicznym kraju powodują konieczność dostosowania profilu wykształcenia absolwenta do potrzeb obecnej praktyki gospodarczej, ale spełnienie tego postulatu wymaga zwiększenia liczby godzin przeznaczonych na dydaktykę i zajęcia praktyczne,
- istnieje potrzeba zwiększenia wyposażenia uczelni w sprzęt komputerowy oraz oprogramowanie, zarówno narzędziowe jak i użytkowe; w planach inwestycyjnych należy zakładać, aby każdy student miał dostęp do komputera,
- należy dokonać międzyuczelnianej wymiany oprogramowania użytkowego dla potrzeb dydaktyki i prac badawczych,
- należy organizować staże w zakresie zastosowań informatyki w firmach menadżerskich,
- nauczanie przedmiotów informatycznych należy dostosować do potrzeb regionu i oczekiwań absolwentów; powinno ono być elastyczne i uwzględniać różnice w przygotowaniu do nowych warunków gospodarki.

Uczestnicy konferencji rozstali się z nadzieją zorganizowania podobnego spotkania za rok lub dwa lata.

JADWIGA ORYLSKA

Zachęcamy wszystkich organizatorów i uczestników konferencji oraz innych imprez o tematyce informatycznej do przesyłania do redakcji swoich refleksji, uwag, sugestii, wniosków. Chętnie je opublikujemy na naszych łamach.

