



ISSN 0239-8044

P3057/2000

2

2000

**Techniki
Komputerowe**
BIULETYN INFORMACYJNY



INSTYTUT MASZYN MATEMATYCZNYCH
WARSZAWA 2000



P3057/2000

Techniki Komputerowe

BIULETYN INFORMACYJNY

Rok XXXV, Nr 2, 2000

INSTYTUT MASZYN MATEMATYCZNYCH
WARSZAWA 2000

0002/30000

Wydaje:

INSTYTUT MASZYN MATEMATYCZNYCH

UL. KRZYWICKIEGO 34

02-798 WARSZAWA

TEL. 621.84.41, TLX 81.78.80, FAX 629.92.70

E-MAIL: imasmat@imm.org.pl

INTERNET: <http://www.imm.org.pl>

Copyright © by Instytut Maszyn Matematycznych, Warszawa 2000

Spis treści

	Str.
Wybrane metody indeksowania i przeszukiwania pełnotekstowego plików w aplikacjach MS Access 97, Małgorzata Rubin	5
Analiza możliwości automatycznej generacji polskich znaków w fontach postscriptowych. Opis programu - dokończenie, Roman Czajkowski, Wojciech Nowakowski, Marcin Paśnikowski	19
Metody symulacji komputerowej i ich przydatność w inżynierii procesów biznesowych, Marek Kacprzak	31
Metody <i>IDEF</i> i ich zastosowanie do <i>BPR</i> , Andrzej Kaczmarczyk	39
Problem efektywności tworzenia oprogramowania w technikach internetowych i propozycja jego rozwiązania, Artur Jakub Jerzak.	49
Internet w kieszeni, czyli wprowadzenie w technikę WAP, Wojciech Kubera	55
Metody rozpoznawania odcisków palców stosowane w systemach kontroli dostępu ze szczególnym uwzględnieniem metod ultradźwiękowych, Wiesław Bicz	61
Optoelektroniczne metody i urządzenia do pomiaru zasięgu widzialności, Romuald Synak	69
8-bitowe mikrosterowniki z pamięcią FLASH programowalne przez synchroniczny interfejs szeregowy SPI, Dariusz Caban	85

MALGORZATA RUBIN

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

Wybrane metody indeksowania i przeszukiwania pełnotekstowego plików w aplikacjach MS Access 97 Selected techniques of indexing and full-text searching the contents of files in MS Access 97 application

Streszczenie

W artykule przedstawiono porównanie wybranych metod indeksowania i przeszukiwania zawartości sformatowanych plików tekstowych oraz możliwość ich wykorzystania w aplikacjach tworzonych w technologii MS Access 97.

Abstract

The article compares the selected techniques of indexing and searching the contents of the formatted text files and describes the methods to apply them in the applications created in MS Access 97 databases technology.

1. Wstęp

O sukcesie w dzisiejszym świecie decyduje coraz częściej czas dostępu do właściwej informacji. Użytkownicy programów wspomagających zarządzanie firmą oczekują od nich sprawnej obsługi nagromadzonych zasobów informacyjnych. Indeksacja i przeszukiwanie pełnotekstowe elektronicznych dokumentów stały się już standardem nowoczesnych aplikacji biznesowych. Bez tego typu mechanizmów i narzędzi efektywność i wydajność wspomaganie zarządzania jest tylko obietnicą bez pokrycia.

Wychodząc naprzeciw oczekiwaniom i potrzebom rynku, w Instytucie Maszyn Matematycznych opracowano system zarządzania dokumentami dokMistrz, zaimplementowany w technologii MS Access 97. Zarządzanie dokumentami obejmuje w nim rejestrację, tworzenie, klasyfikację, przechowywanie, przekaz pomiędzy stanowiskami pracy, dokumentowanie operacji i trasy obiegu dokumentu oraz **wyszukiwanie informacji**.

Dokument w systemie dokMistrz składa się z części opisowej tzw. nagłówka oraz dowolnej liczby podpiętych plików tzw. załączników, w których znajduje się jego właściwa treść. Nagłówek zawiera podstawowe dane, takie jak: nadawca, odbiorca, znak

pisma, temat, czego dotyczy, sprawa, termin załatwienia i inne. Format podpiętych plików może być dowolny (tekst, grafika, dźwięk i inne). Tak zaprojektowana struktura wyznacza w sposób jednoznaczny dwa obszary wyszukiwania dokumentów:

- wyszukiwanie w oparciu o wybrane elementy nagłówka,
- wyszukiwanie według treści załącznika.

Pierwszy rodzaj wyszukiwania zrealizowano w oparciu o filtry, zestawienia oraz operację szukania w ramach wyświetlanej na formularzu listy dokumentów. Filtry i zestawienia umożliwiają wstępne ograniczenie listy dokumentów, natomiast operacja szukania wybiera z niej wszystkie pozycje, spełniające zadane kryterium wyszukiwania. Podstawowym celem stosowania nagłówka jest klasyfikacja dokumentu, czyli powiązanie go z określonym typem, tematem, sprawą itp. W większości są to hasła słownikowe, opisujące szczegółowo każdy dokument i wydają się zawierać wystarczającą informację do wyszukania go. Z drugiej strony, hasła słownikowe raczej systematyzują informację niż nadają jej indywidualne cechy. **Dopiero załącznik w formie pliku jest prawdziwym wyróżnikiem dokumentu.** Wyszukiwanie według treści załącznika pozwala uzyskać bardziej selektywny rezultat na podstawie ogólnego zapytania.

2. Specyfikacja problemu

Proste systemy zarządzania dokumentami nie posiadają opcji pełnotekstowego wyszukiwania dokumentów. Systemy bardziej zaawansowane oferują zarówno wyszukiwanie w oparciu o hierarchiczne klasyfikatory, jak i pełnotekstowe wyszukiwanie dokumentów, wspomagane przez indeksację. Indeksacja jest realizowana przez specjalizowane serwery baz danych jak np. MS SQL Server, czy Lotus Notes/Domino, zapewniające wysoką jakość i wydajność tego procesu.

Klasyfikatory są w istocie hasłami słownikowymi (słowami kluczowymi), tworzącymi tzw. hasłownik. Indeksacja w tym przypadku jest przerzucona na użytkownika. To on musi ręcznie przypisać poszczególnym dokumentom hasła, według których będą one potem wyszukiwane. Dokumenty, które nie zostaną opisane słowami kluczowymi nie będą objęte wyszukiwaniem. Wymaganie od użytkownika przypisywania słów kluczowych podczas rejestracji/edycji dokumentu jest rozwiązaniem niewygodnym i uciążliwym. Doświadczenia z wdrożeń dowodzą istnienia nieustannego problemu odnalezienia się końcowych użytkowników w wirtualnej rzeczywistości, jaką daje program komputerowy oraz wciąż niezadawalającej kultury informatycznej ich samych. Problemem bywa wypełnienie podstawowych pól formularza, a wykorzystanie funkcji aplikacji ograniczone jest do minimum. Z tego powodu należy raczej koncentrować działania na automatyzacji procesów wszędzie tam gdzie to możliwe, ponieważ istnieje większe prawdopodobieństwo wykorzystania ich przez końcowego odbiorcę.

MS Access 97 w swojej bogatej paletce usług nie oferuje gotowych mechanizmów indeksacji plików. Aby sprostać temu wyzwaniu trzeba szukać wsparcia na zewnątrz. W dalszej części artykułu przedstawiono porównanie 3-ech wybranych metod indeksowania i wyszukiwania pełnotekstowego, wykorzystujących odmienne narzędzia i technologie.

3. Indeksacja i wyszukiwanie dokumentów

3.1. Wyszukiwanie dokumentów metodą FileSearch

3.1.1. Model obiektowy

Model obiektowy Microsoft Office zawiera obiekt **FileSearch**, który jest funkcjonalnym odpowiednikiem wspólnego dla wszystkich aplikacji MS Office okna dialogowego *Otwórz*, dostępnego z poziomu menu Plik. Obiekt ten realizuje wyszukiwanie plików według ustalonych kryteriów. W szczególności umożliwia on przeszukiwanie plików określonego typu, zawierającego podany tekst jako treść lub właściwość, w ramach wybranego foldera. Przeszukiwanie odbywa się na bieżąco i nie wymaga uprzedniego indeksowania. Obiekt FileSearch znajduje się w bibliotece MSO97.dll.



Kolekcja PropertyTests reprezentuje kryteria wyszukiwania. Obiekt **FoundFiles** zwraca wynik wyszukiwania w postaci listy pełnych nazw znalezionych plików.

3.1.2. Implementacja

Opcja wyszukiwania metodą FileSearch umożliwia:

- a) wyświetlenie pełnej listy załączników,
Jest to pomocne w sytuacji, kiedy użytkownik nie jest pewien, do których dokumentów podpiął pliki.
- b) wyświetlenie załączników wskazanego typu
Dostępne są następujące typy plików:
 - arkusze Microsoft Excel (*.xls, *.xlw),
 - dokumenty Microsoft Word (*.doc),
 - pliki Microsoft Office (*.doc, *.dot, *.xls, *.xlt, *.htm, *.ppt, *.mdb);
 - pliki RTF (*.rtf),
 - pliki HTM (*.htm),
 - pliki tekstowe (*.txt),
 - pliki bez rozszerzenia (*.),
 - wszystkie (*.*).
- c) wyświetlenie załączników zawierających podany tekst jako treść lub właściwość
Właściwościami pliku nazywamy szczegóły dotyczące pliku, które mogą być pomocne przy jego identyfikacji. Należą do nich opisowy tytuł, autor i temat,

a także słowa kluczowe określające zawartość pliku i inne ważne informacje. Właściwości mogą zostać ustawione dla dowolnego pliku pakietu Microsoft Office. Do standardowych właściwości pliku należą: autor, tytuł i temat pliku. Na przykład, pracując z programem Word, po wstawieniu do każdego pliku związanego ze sprzedażą słowa kluczowego „PZ” możliwe jest późniejsze odnajdywanie tych plików przez wyszukiwanie według tego słowa kluczowego. Możliwe jest także tworzenie niestandardowych właściwości pliku. Aby przy pierwszym zapisie każdego pliku automatycznie pojawiała się okno dialogowe *Właściwości*, należy z menu *Narzędzia* wybrać polecenie *Opcje*, kliknąć kartę *Zapisywanie* i zaznaczyć pole wyboru *Żądanie podawania właściwości dokumentu*.

d) wyświetlenie załączników na podstawie daty ich modyfikacji

Zestaw dostępnych przedziałów czasowych obejmuje: datę bieżącą, bieżący tydzień, ostatni tydzień, bieżący miesiąc, ostatni miesiąc, kiedykolwiek.

Znajdź w załączniku

typ załącznika: plik Microsoft Office tekst lub właściwość: dokm* zmieniajany: kiedykolwiek

Znak pisma	Typ	Kontrahent	Nr systemowy	Nr zał	Opis załącznika	Rodzaj	Nazwa źródłowa	
RF11/99	P	pismo	"Optim" Sp. z o.o.	2000/01/00014	30	sprowozdanie dokmistrza	doc	DModbio2000
ZAL/Unicode	P		Towarzystwo Turystyczne	2000/04/00043	01	dokument 2-stronicowy	doc	2_strony
machewka	W		PUH Tramp	2000/04/00045	01	dokmistrz w arkuszu	xls	
OPT/F/02/00	P	faktura	"Optim" Sp. z o.o.	2000/04/00053	01	dokument 1-stronicowy	doc	1_strona
wych_new	W	faktura	PUH Tramp	2000/06/00072	03	tekst o dokmistrzu	doc	
radość	W	adnotacje urzęd.	Przedsiębiorstwo Usługowe	2000/06/00074	02	podręcznik	doc	Kopia podr14
2000wyrazów	P	kalkulacja	Wywórnia Sprzętu	2000/06/00084	01	plik zawierający 2000 wyre	doc	DModbio/2000
baza mdb	P	faktura	Microsoft	2000/06/00087	01	baza mdb	mdb	dokmisprz

Liczba znalezionych załączników: 8 zakres: [] [Otwórz] [Zamknij]

Pole wypełniane treścią lub właściwością załącznika akceptuje symbole wieloznaczne. Pozwala to wyszukiwać słowa niezależnie od ich przedrostków i końcówek gramatycznych. Znak ? zastępuje dowolny, pojedynczy znak w słowie, a znak * zastępuje dowolny ciąg znaków. Wpisanie obok siebie dwóch słów oddzielonych spacją nie oznacza, że będzie poszukiwana fraza. Przy wyszukiwaniu nieistotne jest położenie wyrazów w tekście. Wybrane zostaną te pliki, które zawierają wszystkie wpisane wyrazy, przy czym treść i właściwości są traktowane łącznie. Rezultat wyszukania ciągu znaków: „instrukcja dokMistrz”, wpisanego w polu *tekst lub właściwość* obejmie załączniki, które:

- zawierają słowa „instrukcja” i „dokMistrz” w dowolnym miejscu swojej treści,
- zawierają słowa „instrukcja” i „dokMistrz” w swoich właściwościach,
- zawierają jedno ze słów w swojej treści, a drugie jako właściwość.

3.1.3. Wnioski

Zaletą wyszukiwania metodą FileSearch jest niewątpliwie to, że nie wymaga ona indeksowania, ani też instalacji dodatkowego oprogramowania. W zamian za to, wyszukiwanie nie należy do najszybszych, a czas oczekiwania na wynik zależy od stopnia precyzji kryterium wyszukiwania oraz objętości przeszukiwanego foldera plików. Im więcej zadanych elementów, tym wyszukiwanie jest wolniejsze. Wpływ poszczególnych elementów kryterium na wydłużenie procesu wyszukiwania nie jest jednakowy. Najbardziej krytyczne pod tym względem pozostaje pole tekstowe, określające *tekst lub właściwość*. Z tego względu, wyszukiwanie metodą FileSearch nie powinno być nadużywane, lecz stosowane „oszczędnie”. Dodatkową niedogodnością jest brak możliwości rozdzielenia treści i właściwości. Chociaż w większości przypadków poszukiwany ciąg znaków dotyczy treści, zwracany rezultat jest zawsze łączny i obejmuje załączniki spełniające kryterium zarówno pod względem treści jak i właściwości. Programowe uzyskanie dodatkowych informacji nie jest możliwe, ponieważ obiekt FoundFiles zwraca jedynie nazwy znalezionych plików.

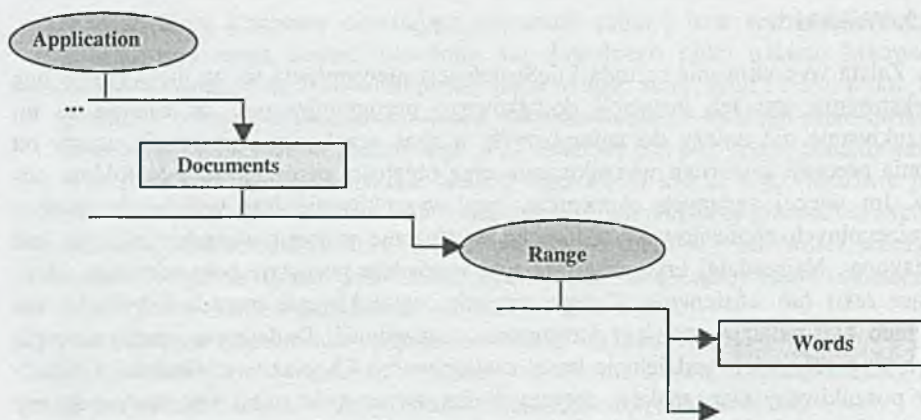
Rzetelnie wypełniane, przynajmniej niektóre, właściwości dokumentów mogą stanowić hasłownik (zbiór słów kluczowych), według którego możliwe będzie potem ich odnalezienie.

3.2. Indeksacja w oparciu o MS Word

Indeks jest elementem, który niewątpliwie wspomaga proces wyszukiwania pełnotekstowego. Najprostsza struktura indeksu zawiera listę unikalnych słów kluczowych i ich powiązań z odpowiednimi dokumentami. Integracja aplikacji w ramach pakietu MS Office, w tym spójny model obiektowy, stwarza możliwość swobodnego wykorzystania specyficznych cech i funkcji obiektów. Dokumenty tworzone przy pomocy różnych edytorów tekstu są najczęściej plikami, zawierającymi sformatowany tekst. Ze względu na znaki sterujące, obróbka takich plików nie jest problemem trywialnym i wymaga zaawansowanych algorytmów analizy tekstu. Jeżeli jednak treść dokumentów jest tworzona przy użyciu edytora MS Word (a takie pliki są najbardziej popularne), to można wykorzystać technikę OLE (Object Linking and Embedding) do uzyskania listy słów w nich zawartych i na tej podstawie podjąć próbę utworzenia własnego indeksu.

3.2.1. Model obiektowy

Model obiektowy Microsoft Word zawiera kolekcję **Documents**, obejmującą wszystkie otwarte dokumenty w MS Word. Każdy obiekt **Document**, należący do kolekcji **Documents**, posiada własną kolekcję **Words**, składającą się ze wszystkich słów znajdujących się w ramach określonego obszaru **Range**.

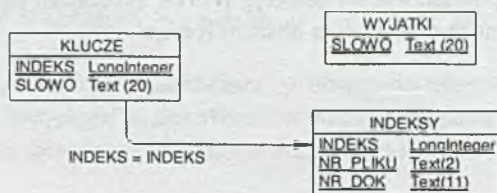


Biblioteka obiektów MS Word została zawarta w pliku o nazwie MSWORD8.OLB

3.2.2. Model danych

Struktura indeksu została oparta na trzech tabelach: *klucze*, *wyjątki* oraz *indeksy*. Pierwsza z nich jest zbiorem unikalnych słów kluczowych (hasel), wyłonionych w procesie analizy treści poszczególnych załączników. Zbiór ten nie obejmuje słów nieistotnych znaczeniowo, takich jak: i, oraz, pod, nad, też, lub, także, ja ty, on, ... Słowa tego typu znajdują się w tabeli *wyjątki*. Wynik indeksacji jest zapisywany do tabeli *indeksy*. Dołączanie i tworzenie nowych załączników dokumentów wpływa na dezaktualizację raz utworzonego indeksu. Można uznać, że w większości przypadków edycja raz zredagowanego załącznika nie wnosi „drastycznych” zmian do jego warstwy znaczeniowej. Jeśli zatem był on poddany indeksacji – to zbiór podstawowych słów kluczowych pozostaje w zasadzie niezmienny. Inaczej jest w przypadku tworzenia nowych plików, nie objętych indeksem. Dopóki indeks nie zostanie uaktualniony dokumenty te nie będą widoczne w trakcie operacji wyszukiwania. W związku z powyższym, proces indeksacji musi być okresowo ponawiany, przy czym przetwarzane powinny być tylko pliki modyfikowane lub nowoutworzone od daty ostatniej generacji indeksu. Aktualizacja może być przeprowadzana okresowo, bądź na bieżąco, po zakończeniu edycji lub tworzenia załącznika dokumentu.

Wyszukiwanie dokumentów w oparciu o istniejący indeks polega na wybraniu zestawu rekordów z tabeli *indeksy*, opatrzonych identyfikatorem zadanego słowa kluczowego.



3.2.3. Implementacja

Przy implementacji algorytmu indeksacji wykorzystano technikę OLE do otwierania poszczególnych plików w tle, jako dokumenty MS Word, w celu pobrania elementów kolekcji Words każdego z nich. Na tej podstawie uzyskano zestaw słów kluczowych, potrzebny do zbudowania indeksu dokumentów.

Przeprowadzono testy wydajnościowe algorytmu indeksacji dla plików o różnej objętości tekstu.

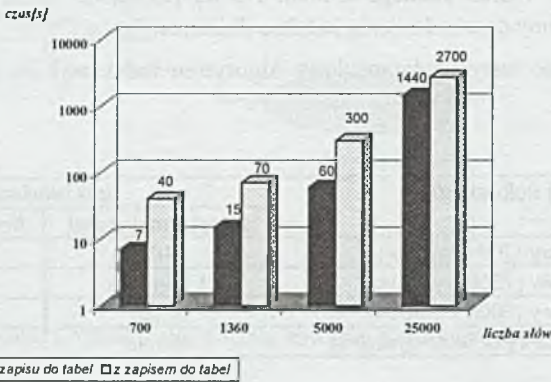
rodzaj dokumentu	czas indeksacji	
	z zapisem do tabel	bez zapisu do tabel
dokument 1 stronicowy (700 wyrazów)	40s	7 s
dokument 2 stronicowy (1360 wyrazów)	1 min 10 s	15 s
dokument 8 stronicowy (5000 wyrazów)	5 min	1 min
dokument 100 stronicowy (25000 wyrazów)	45 min	24 min

Otrzymane wyniki świadczą o małej wydajności zarówno techniki OLE (długi czas pobierania słów z kolekcji) jak i samego dostępu do tabel przez MS Access. W celu optymalizacji czasu indeksowania zmodyfikowano algorytm w części dotyczącej analizy tekstu. MS Word wykorzystano tylko do konwersji plików zawierających tekst sformatowany na zwykłe pliki tekstowe, aby te z kolei samodzielnie przetwarzać i dzielić na słowa.

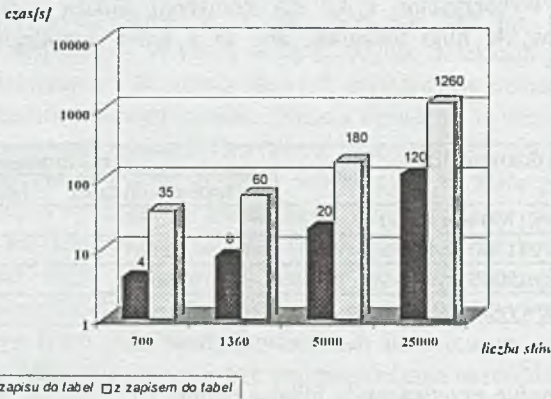
rodzaj dokumentu	czas indeksacji	
	z zapisem do tabel	bez zapisu do tabel
dokument 1 stronicowy (700 wyrazów)	35s	4 s
dokument 2 stronicowy (1360 wyrazów)	1 min	8 s
dokument 8 stronicowy (5000 wyrazów)	3 min	20 s
dokument 100 stronicowy (25000 wyrazów)	21 min	2 min

Przy takim sposobie przetwarzania plików poprawa wydajności jest znacząca dla plików o większej objętości tekstu. Tempo indeksacji małych jest porównywalne przy zastosowaniu obydwu algorytmów. Czas potrzebny na konwersję pliku na tekstowy, a następnie jego zapis na dysk kompensuje opóźnienie związane z pobieraniem słów z kolekcji Words. Konwersja plików, a tym samym optymalizacja indeksacji jest więc opłacalna dla plików zawierających powyżej 5000 słów. Po otwarciu dokumentu w MS Word należałoby więc najpierw sprawdzać liczbę słów (poprzez odczyt właściwości *Count*) i dopiero na tej podstawie podejmować decyzję o ewentualnej konwersji tego pliku.

Indeksacja oparta na kolekcji Words



Indeksacja z konwersją na plik tekstowy



3.2.4. Wnioski

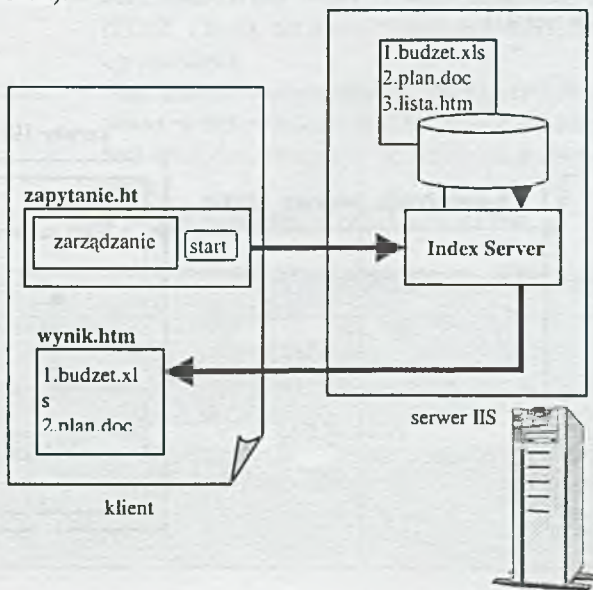
Indeksowanie plików z wykorzystaniem MS Word i techniki OLE jest operacją czasochłonną i mało wydajną. Optymalizacja tego procesu dotyczy jedynie przetwarzania i analizy tekstu. Skrócenie czasu dostępu do tabel w MS Access nie jest możliwe. Indeksacja w takim wydaniu na pewno nie przyczyni się do podniesienia jakości i konkurencyjności aplikacji jako produktu rynkowego. Czas potrzebny na indeksację nawet małego pliku jest zbyt duży na to, aby można było ją stosować na bieżąco. Okresowa aktualizacja indeksu jest natomiast kłopotliwa ze względu na małą wydajność operacji, a ponadto wymaga systematyczności, w przeciwnym razie opcja wyszukiwania dokumentów traci sens. Indeksacja tego typu może okazać się przydatną np. przy tworzeniu archiwum dokumentów, kiedy dysponujemy kompletem dokumentów i mamy pewność,

że nie będą one już więcej modyfikowane. Można wtedy dopuścić jednorazową, nawet długotrwałą indeksację, aby wykorzystać potem jej wyniki do wyszukiwania dokumentów w archiwum.

3.3. Indeksacja w oparciu o Microsoft Index Server

Na rynku oprogramowania istnieje produkt firmy Microsoft – Index Server, służący do indeksowania i wyszukiwania pełnotekstowego dokumentów HTML, tradycyjnych plików tekstowych oraz dokumentów sformatowanych MS Office. Stworzono go z myślą o indeksowaniu głównie zasobów w sieci WWW, ale jego uniwersalizm w zakresie rozpoznawanych formatów plików sprawia, że może być on z powodzeniem stosowany do przeszukiwania zasobów dokumentów w firmowym intranecie. Funkcjonalność Index Server-a jest idealnym rozwiązaniem dla systemów zarządzania dokumentami w zakresie ich wyszukiwania. Dynamiczne wypieranie tradycyjnych metod programowania przez technologię WWW może być dodatkowym atutem w uzasadnieniu decyzji o zastosowaniu właśnie Index Server-a.

Index Server jest pakietem zintegrowanym z Internet Information Server 4.0 (IIS) i rozpowszechnianym w ramach Windows NT Option Pack 4.0. Dzięki temu, jego instalacja jest prosta i w pełni automatyczna. Charakteryzuje się on dużą wydajnością i niezawodnością. Samo indeksowanie obejmuje zarówno treść jak i właściwości dokumentu (różne dla różnych typów plików). Monitorowanie dokumentów jest procesem ciągłym. Dzięki wykorzystaniu systemowych mechanizmów powiadamiania o zmianach w skatalogowanych dokumentach, Index Server nie musi za każdym razem przeglądać pełnej zawartości systemu plików w celu sprawdzenia, który z przetworzonych wcześniej plików został od tego czasu zmieniony i należy w związku z tym uaktualnić jego indeks. Operacja przejrzania całej struktury odbywa się tylko raz – w momencie tworzenia indeksu (initial scan). Nowo dodane i modyfikowane pliki wykrywane są w trakcie wykonywanego co pewien czas skanowania przyrostowego (incremental scan).



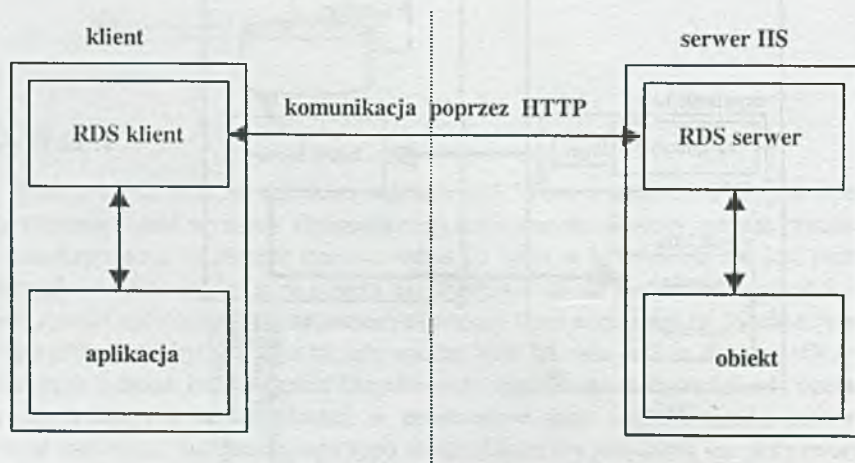
Index Server oferuje metodę odpytywania, zwaną w tym przypadku językiem zapytań (query language). Umożliwia to zadawanie nawet bardzo skomplikowanych zapytań, obejmujących kryteriami każdy z atrybutów poindeksowanych plików. Wykorzystanie zalet Index Server-a z poziomu aplikacji jest możliwe poprzez interfejs OLE DB. Dzięki niemu można wykonywać zapytania o składni zbliżonej do języka SQL, a poprzez technologię ADO (ActiveX Data Objects) pobierać uzyskane dane.

Ze względu na ścisłą integrację Index Server-a z IIS (Internet Information Server), jego obsługa, a w szczególności włączanie i wyłączanie indeksowania odbywa się z poziomu aplikacji zarządzającej IIS.

3.3.1. Index Server, a technologia tradycyjna

Wysoka jakość indeksacji i wyszukiwania musi być okupiona określonymi wymaganiami sprzętowymi i programowymi. Fakt ten jest niekwestionowany i oczywisty. Index Server powstał z myślą o aplikacjach internetowych, które w swej naturze obejmują wymianę danych pomiędzy przeglądarką - klientem, a serwerem WWW. Interpretery języków skryptowych potrafią odróżnić część kodu, która przeznaczona jest do wykonania na serwerze, od elementów wykonywanych po stronie klienta. Przez to klient może uruchamiać proces na serwerze i dostaje gotowy wynik, który może wyświetlić. Sytuacja komplikuje się, gdy mamy do czynienia z aplikacjami wykonanymi w technologii tradycyjnej, opartych np. na MS Access 97 i języku VBA. Pojawia się problem: jak uzyskać dostęp do zdalnego obiektu, pracującego na serwerze. Rozwiązaniem w takim przypadku jest technologia opracowana przez firmę Microsoft określana jako RDS (Remote Data Service). Pozwala ona na komunikowanie się aplikacji pracującej po stronie klienta z obiektem (business object) uruchomionym na serwerze. Metoda ta udostępnia wskazanie na obiekt, umożliwiające pośrednie korzystanie z jego metod.

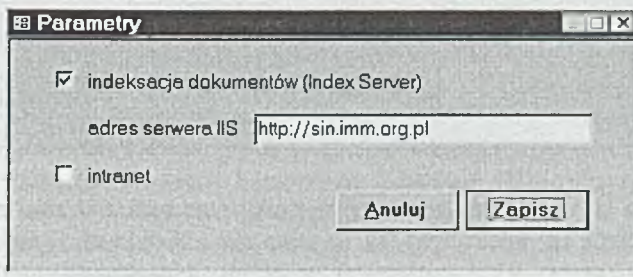
Komponenty po stronie serwera instalowane są automatycznie w trakcie instalacji IIS (typowej lub minimalnej), natomiast po stronie klienta – są składową programu Microsoft Internet Explorer 4.0.



Wykorzystanie zaawansowanej technologii RDS wymaga wykroczenia poza środowisko programistyczne MS Access 97. Przy pomocy np. Microsoft Visual Basic 5.0, trzeba zdefiniować klasę obiektu, powoływanego i uruchamianego na serwerze z poziomu aplikacji tradycyjnej. W naszym przypadku klasa obiektu została nazwana **DOKMISTRZ.search** i zapisana w pliku **dokmistrz.dll**, umieszczonym na serwerze. Do zadań obiektu klasy **DOKMISTRZ.search** należy komunikowanie się poprzez ADO z Index Server-em, a w szczególności: uruchamianie metody **Execute** dla zapytań SQL, pobieranie wyniku w postaci obiektu **Recordset** (wybranego zestawu rekordów spełniających kryterium) i zapisywanie go poprzez ADO i ODBC w bazie danych MS Access. Stamtąd wynik jest już odczytywany i wyświetlany na formularzu aplikacji klienta.

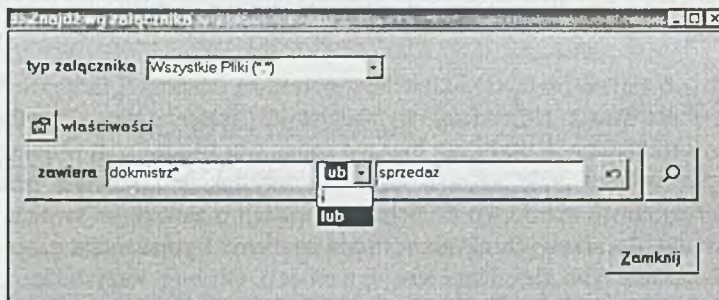
3.3.2. Implementacja

Korzystanie z Index Server-a wymaga określenia w aplikacji *adresu serwera IIS*, niezbędnego także do poprawnej pracy RDS.

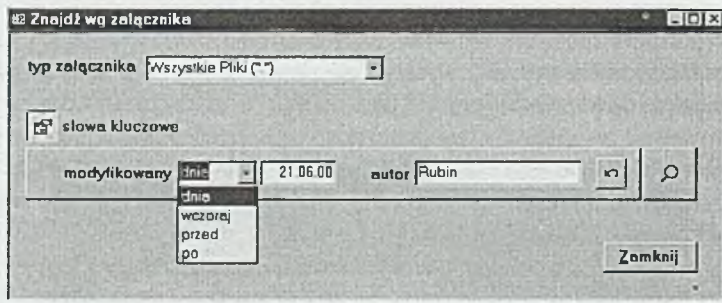


Proste zapytanie zawiera następujące elementy:

- a) typ załącznika – ograniczenie przeszukiwania do wybranych formatów plików: Microsoft Word (*.doc, .dot), Microsoft Excel (.xls, .xlw), HTML (.htm), tekstowe (*.txt) lub wszystkie (*.*), czyli brak ograniczenia;
- b) słowo lub fraza – ciąg znaków interpretowany jako poszukiwana fraza tekstu lub słowo w treści dokumentu; może być zapisany za pomocą symboli wieloznacznych (*, ?); pojedyncze litery i cyfry są ignorowane
- c) relacja – wybrano najprostsze typy relacji: **i / lub**.



Spośród wielu właściwości zostały wybrane tylko dwie, uznane za najbardziej przydatne i nie wymagające żadnych dodatkowych działań ze strony użytkownika, związane z *datą modyfikacji* pliku oraz jego *autorem*. Pozwalają one wyszukać np. pliki własne, modyfikowane danego dnia, wczoraj, przed i po określonym dniu. Data modyfikacji jest przechowywana w strukturach Index Server-a w formacie długim co oznacza, że przy konstruowaniu zapytania należy uwzględniać także czas, w przeciwnym razie generowany jest błąd składniowy wyrażenia SQL. Szczegółowe dane o właściwościach dokumentu, przechowywane i zwracane przez Index Server są dostępne z poziomu aplikacji i można je wyświetlić na formularzu. Nie było to możliwe w metodzie FileSearch.



Początkowa koncepcja implementacji wyszukiwania była związana z jednym formularzem (podobnie jak FileSearch) tzn. to samo okno służyło zarówno do określenia kryterium wyszukiwania jak i prezentacji wyników otrzymanych z Index Server-a. W trakcie prowadzonych prac okazało się, że metoda Requery uruchamiana dla formularza w MS Access 97 nie działa właściwie. W naszym, konkretnym przypadku formularz był formularzem związanym ze zdefiniowanym źródłem danych w postaci kwerendy opartej na tabeli *wyniki*. Dane do tej tabeli były wstawiane przez zewnętrzny obiekt klasy DOKMISTRZ.search, a nie przez MS Access. Próba ich wyświetlenia z poziomu MS Access kończyła się przypadkowym rezultatem. Niekiedy lista wynikowa była pusta, mimo istnienia fizycznych danych w tabeli. Przyczyna błędu być może tkwi w buforowaniu danych, wstawianych do tabel MS Access przez obiekt zewnętrzny, które z tego powodu mogą być przez pewien czas niedostępne, a tym samym nie objęte zapytaniem. Powołanie odrębnego formularza do prezentacji wyników wyszukiwania rozwiązywało problem. Przy otwieraniu formularza MS Access już prawidłowo obsługiwał źródło danych i błąd nie występował.

3.3.3. Wnioski

Indeksacja i wyszukiwanie dokumentów, oparte na Microsoft Index Server, spełniają oczekiwania nawet najbardziej wymagających projektantów i użytkowników. Profesjonalny mechanizm śledzenia na bieżąco zmian wprowadzanych w dokumentach i natychmiastowe uaktualnianie indeksu zapewnia satysfakcję i komfort pracy. Użytkownik w każdej chwili ma dostęp do pełnej informacji o zawartości swoich zasobów, bez wykonywania dodatkowych czynności (uruchamiania i ponawiania procesu indeksacji oraz obciążania tymi zadaniami swojej pamięci). Obsługę wszystkiego przejmuje na siebie Index Server. Obszerna informacja, zawarta w zasobach indeksu oraz wyraźne

rozdzielenie treści od właściwości pozwala użytkownikowi uzyskiwać selektywne wyniki, w zależności od interesujących go elementów. Ponadto, zwracany rezultat może być wyświetlony na ekranie w szczegółowej formie, zawierającej wartości wybranych właściwości znalezionych plików. Będą one zawsze widoczne niezależnie od tego, czy zostały określone w kryterium wyszukiwania, czy też nie, a przez to pomocne w identyfikacji załącznika. Wyszukiwane mogą być zarówno pojedyncze słowa jak i pełne frazy tekstu.

Wysoka jakość i wydajność operacji wyszukiwania jest okupiona jedynie koniecznością posiadania serwera sieciowego Windows NT, natomiast oprogramowanie Windows NT 4.0. Option Pack, zawierające IIS oraz Index Server jest bezpłatnie licencjonowane i udostępnione na stronie WWW firmy Microsoft

4. Podsumowanie

Istnieje wiele rozwiązań problemu indeksacji, ale powyższe przybliżenie wybranych metod ma aspekt praktyczny dla czytelników. Twórcy oprogramowania w MS Access 97 nie mają w tej dziedzinie ułatwionego zadania. Wady i zalety poszczególnych metod dowodzą, jak różne mogą być obszary ich użycia oraz jakie wiążą się z tym problemy. Klasa konkretnej aplikacji i skala jej zastosowań wymusza wykorzystanie odpowiednio dobranej indeksacji pod względem jakości i wydajności. W całej problematyce rozważań na ten temat bezspornym pozostaje fakt, że przy obecnym upowszechnieniu specjalizowanych narzędzi indeksacji, ambicjonalne traktowanie tego zagadnienia poprzez generowanie i obsługę indeksu „własnymi siłami” jest przedsięwzięciem całkowicie nieopłacalnym.

Literatura

- [1] „*Microsoft Visual Basic for Applications and Shared Libraries References*”, Microsoft Press 1997.
- [2] „*Microsoft Word Language Reference*”, Microsoft Press 1997.
- [3] „*Microsoft Excel Language Reference*”, Microsoft Press 1997.
- [4] „*Programming ADO*”, Microsoft Press 2000.
- [5] „*Access 97 Biblia*”, Cary N. Prague, Michael R. Irwin, IDG Books Worldwide, Inc 1997.
- [6] „*Using Internet Information Server 4*”, Nelson Howell, Que Corporation 1997.
- [7] *Microsoft Support Knowledge Base*, www.microsoft.com.
- [8] *Microsoft Office 97 Developer Edition Tools Sampler Documentation*.
- [9] *Windows NT Option Pack Documentation*.
- [10] *MSDN Library – The Essential Resource for Developers*.

ROMAN CZAJKOWSKI, WOJCIECH NOWAKOWSKI
INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA
MARCIN PAŚNIKOWSKI
POLTECHNIKA WARSZAWSKA

Analiza możliwości automatycznej generacji polskich znaków w fontach postscriptowych
Opis programu – dokończenie
An analysis of the possibilities of the automatic generation of Polish fonts in the Postscript format
A description of the program – final part

Streszczenie

W artykule omówiono moduły bloku edycji, opisano sposób obsługi programu i przedstawiono wnioski podsumowujące wszystkie części tej pracy.

Abstract

The paper describes the modules of the edition block, the program usage and gives the final conclusions.

1.5. Blok zapisu

Ostatnim blokiem, którego funkcje wiążą się z przetwarzaniem danych wejściowych jest blok edycji. Po zakończeniu jego działalności blok danych, pełniący rolę bazy danych zawiera komplet informacji w oparciu o które można wygenerować nową spolonizowaną czcionkę. Za proces generacji nowego fontu odpowiedzialny jest ostatni człon programu – blok zapisu. Jego funkcje ograniczają się wyłącznie do operacji przekształcających dane zawarte w bloku danych do postaci zgodnych z formatami plików fontowych standardu Type 1. Tak jak w przypadku pozostałych bloków, blok zapisu jest podzielony na funkcjonalne moduły, z których każdy realizuje pewne wydzielone operacje. Moduły te zostały zorganizowane w postaci trzech gałęzi, z których każda generuje dane wchodzące w skład innego pliku fontowego. Gałąź pierwsza (patrz rys. 4, [2]) składa się z dwóch modułów: modułu generacji ciała fontu i modułu szyfrowania ciała fontu. Ich produktem wyjściowym jest plik *.pfb, czyli zbiór danych

w formacie Postscript Font Body. W skład gałęzi pierwszej wchodzi również opcjonalny moduł formatowania ciała fontu. Przekształca on zaszyfrowany font do formatu Postscript Font ASCII, który następnie zostaje zapisany w zbiorze o rozszerzeniu *.pfa.

Galęzie druga i trzecia, zawierająca odpowiednio moduł generacji zbioru *.pfm i *.afm odpowiedzialne są za generowanie zbiorów metrycznych: *Printer Font Metrics* i *ASCII Font Metrix*. Zbiory te zapisywane są w postaci plików o rozszerzeniu *.pfm i *.afm.

1.5.1. Moduł generacji ciała fontu

Aby omówić zasadę działania modułu generacji ciała fontu należy na moment powrócić do rozdziału 1.2.2 [2] opisującego moduł analizy, ponieważ moduł ten pełni funkcje analogiczne do obecnie omawianego modułu generacji ciała fontu. Zadaniem modułu analizy była interpretacja danych wejściowych w postaci rozszyfrowanego zbioru *.pfb. Moduł ten skanował zbiór wejściowy w poszukiwaniu istotnych danych z punktu widzenia celu programu. Gdy dane takie zostały znalezione, były one przenieszone do bloku danych. Jako rezultat tych operacji otrzymywano blok danych wypełniony istotnymi informacjami oraz plik wyjściowy o znacznie zmniejszonym rozmiarze w stosunku do pliku wejściowego. Plik ten zawierał dane nie wykorzystywane przez żaden inny moduł programu.

Moduł generacji ciała fontu pełni funkcje odwrotne względem modułu analizy. Jego zadanie polega na odtworzeniu pełnej zawartości zbioru *.pfb poprzez uzupełnienie fragmentów usuniętych przez moduł analizy, danymi zawartymi w bloku danych. W momencie zapisu, a więc w czasie działania modułu generacji ciała fontu, blok danych zawiera już informacje na temat polskich znaków. Na wyjściu otrzymuje się kompletny zbiór danych opisujących czcionkę w postaci nie zakodowanego pliku *.pfb.

1.5.2. Moduł szyfrowania ciała fontu

Ostatnią operacją wieńczącą proces przetwarzania zbioru *.pfb jest szyfrowanie. Jest ono realizowane przez moduł szyfrowania ciała fontu, który pracuje podobnie jak opisany w rozdziale 1.2.1 [2] moduł deszyfrowania. Inna jest tylko kolejność wykonywanych działań. Algorytm szyfrujący jest analogiczny do algorytmu deszyfrującego i przedstawia się następująco:

1. Należy dodać na początek tekstu przeznaczonego do zaszyfrowania liczbę #n bajtów randomicznych o dowolnej wartości, gdzie #n jest zazwyczaj równe 4 (w innych przypadkach jest określone przez pozycję słownika **Private** o nazwie **lenIV**).
2. Należy zainicjować klucz kodujący **R** wartością 55665, czyli taką samą jak w przypadku algorytmu deszyfrującego.
3. Następnie należy, począwszy od czterech pierwszych randomicznych bajtów, brać kolejne 8-bitowe znaki tekstu (dla ustalenia uwagi określane tu literą **P**) i wykonać na nich następujące czynności:
 - a) przypisać pomocniczej zmiennej **T** osiem starszych bitów klucza kodującego **R**;

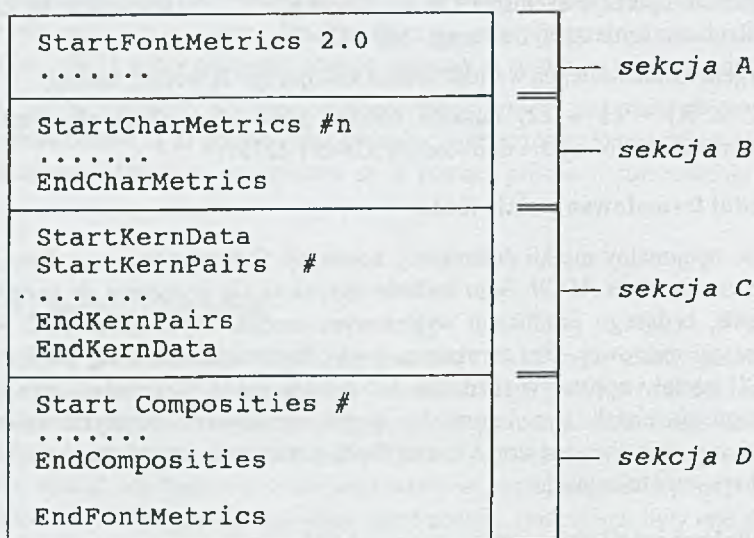
- b) wykonać operację exclusive-OR na zmiennych T i P; rezultatem tej czynności będzie uzyskanie zaszyfrowanego bajtu C.
- c) wygenerować następną wartość klucza kodującego R według formuły:
 $((C + R) * c1 + c2) \text{ modulo } 65536$, gdzie c1 i c2 są pewnymi stałymi o wartościach równych odpowiednio 52845 i 22719;

1.5.3. Moduł formatowania ciała fontu

Jest to opcjonalny moduł dokonujący konwersji formatu *Postscript Font Body* na format *Postscript Font ASCII*. Jego zadanie ogranicza się wyłącznie do przetwarzania zbioru *.pfb, będącego produktem wyjściowym modułu szyfrowania ciała fontu, na zbiór o postaci tekstowej. Oba formaty zarówno Postscript Font Body jak i Postscript Font ASCII zostały opisane w rozdziale 1.2 poświęconym blokowi odczytu. Na podstawie przedstawionych tam informacji został opracowany algorytm formatujący. Identyfikuje on zaszyfrowaną część zbioru *.pfb a następnie przekształca poszczególne jej bajty do postaci tekstowej.

1.5.4. Moduł generacji zbioru metrycznego AFM

Projektowana aplikacja potrafi już wygenerować zmodyfikowany zbiór *.pfb czyli Postscript Font Body. Mimo, iż zawiera on praktycznie wszystkie możliwe dane na temat czcionki (oprócz kerningu), to jednak w praktyce dysponowanie wyłącznie zbiorem *.pfb jest niewystarczające. Współcześnie, większość programów przygotowujące dokumenty do druku pracują w technologii WYSIWYG. Termin ten jest skrótem od słów What You See Is What You Get i oznacza tyle co możliwość edycji dokumentu w takiej formie, w jakiej będzie prezentować się na wydruku. Dzięki temu można być pewnym co do ostatecznego wyglądu dokumentu, już w fazie jego projektowania. Technologia ta wymaga jednak pewnych danych, na podstawie których program pracujący w technologii WYSIWYG mógłby wygenerować wygląd wydrukowanego dokumentu. Aby móc we właściwy sposób rozmieścić tekst na dokumencie, trzeba dysponować zbiorem pewnych wartości liczbowych opisujących poszczególne znaki. Oczywiście można by wydobyć je z pliku *.pfb, lecz wymagałoby to od aplikacji stosowania dość zaawansowanych technik i wiązałoby się z dużymi nakładami obliczeniowymi i czasowymi. Aby nie obciążać tymi zadaniami programów składających teksty, pewna część użytecznych danych jest wyekstrahowana ze zbioru *.pfb i zapisana w specjalnie do tego celu przeznaczonym zbiorze metrycznym. Zbiór ten jest określany mianem Adobe Font Metrics i posiada rozszerzenie *.afm. Charakteryzuje się ściśle określoną strukturą danych, zapisanych w formacie tekstowym. Jego ogólna struktura przedstawia się następująco:



Rys. 18. Format zbioru Adobe Font Metrics

W sekcji A zawarte są informacje ogólne opisujące font jako całość. Poniżej znajduje się przykład tej części zbioru metrycznego:

StartFontMetrics 3.0

Comment Copyright 1987-1992 as an unpublished work by Bitstream Inc.

FontName Swiss721BT-Light

FullName Swiss 721 Light

FamilyName Swiss 721

Weight Light

ItalicAngle 0

IsFixedPitch false

FontBBox -167 -236 1181 965

UnderlinePosition -104

UnderlineThickness 49

Version 003.001

Notice Copyright 1987-1992 as an unpublished work by Bitstream Inc.

EncodingScheme AdobeStandardEncoding

CapHeight 719

XHeight 519

Ascender 719

Descender -209

Sekcja A zbioru *.afm zawiera dane zorganizowane w postaci pojedynczych linii tekstu. Te z nich, które rozpoczynają się od słowa *Comment*, interpretowane są jako

komentarz i nie mają większego znaczenia z punktu widzenia programów analizujących zbiory *.afm. Pozostałe linie, poza paroma wyjątkami, definiują zawartość odpowiednich pozycji słownika *FontInfo*. Cztery ostatnie pozycje, tj. **CapHeight**, **XHeight**, **Ascender**, **Descender** nie posiadają swoich odpowiedników w słowniku *FontInfo*, a ich wartości wynikają z pośrednio z danych zawartych w słowniku *CharStrings*. Są to cztery miary pionowe określające wysokości do których docierają pewne grupy znaków o „płaskich zakończeniach”:

CapHeight – określa wysokość do której docierają wielkie litery takie jak X, Z, Y, T, W itp.

XHeight – określa wysokość do której docierają małe litery takie jak x, z, w itp.

Ascender – określa wysokość do której dociera np. litera b

Descender – określa wysokość na której kończy się np. pionowy element litery q.



Rys. 19. CapHeight, XHeight, Ascender, Descender

Linia *StartCharMetrics #n* rozpoczyna sekcję B, określającą wymiary znaków. Linia ta składa się z liczby *#n* powtarzających się sekwencji, z których każda opisuje pojedynczy znak i posiada następujący format:

C #c ; WX #wx ; N [name] ; B #b1 #b2 #b3 #b4 ;

Sekcja składa się z czterech grup oddzielonych od siebie znakami średnika. Grupa pierwsza określa numer kodu pod jakim znak jest widoczny na zewnątrz fontu. Mówiąc inaczej parametr *#c* określa pozycję pod jaką znak występuje w wektorze kodowym. Jeśli *#c* przyjmuje wartość -1 , oznacza to, że dany znak nie występuje w wektorze kodowym.

Grupa druga definiuje długość znaku. Parametr *#wx* jest tu liczbą całkowitą wyrażoną w jednostkach układu współrzędnych *character space*.

Wreszcie, parametr *[name]* w grupie trzeciej określa nazwę znaku. Dokładniej mówią jest to nazwa pod jaką dany *outline* występuje wewnątrz słownika *CharStrings*.

Zadaniem ostatniej grupy rozpoczynającej się literą **B** jest definicja ramki, wewnątrz której mieści się znak. Parametry *#b1 #b2 #b3 #b4* określają współrzędną lewego dolnego i prawego górnego wierzchołka prostokąta opisanego na znaku. Poniżej została przedstawiona definicja znaku dolara:

C 36 ; WX 532 : N dollar ; B 29 -98 504 759 ;

W trzeciej z kolei sekcji **C** zawarte są informacje na temat kerningu. Kerning jest to mechanizm, którego zadaniem jest poprawa estetyki odbioru tekstu poprzez wymuszenie zmiany odległości pomiędzy pewnymi parami liter. Okazuje się bowiem, że w pewnych sytuacjach można osiągnąć znacznie lepszy efekt wizualny, gdy sąsiednie znaki zbliży się lub rozsunie na odległość różną od wynikającej z ich faktycznej długości. Za przykład może posłużyć para wielkich liter **L** i **T**. Umieszczone obok siebie wyglądają znacznie lepiej, gdy są zsunięte, tak by się nawzajem „zazębiały”. Takich par znaków jest znacznie więcej i to właśnie one są opisane w sekcji **C**. Opis pojedynczej pary przedstawia się następująco:

KPX [znak1] [znak2] offset

[znak1] [znak2] są to nazwy odpowiednio pierwszego i drugiego znaku tworzącego parę dla której definiowany jest *kerning*. Parametr *offset* jest liczbą całkowitą określającą przesunięcie *[znaku2]* względem pozycji wynikającej z długości *[znaku1]*. Parametr ten jest wyrażany we współrzędnych układu *character space*. Kiedy przyjmuje wartości ujemne oznacza to, że *[znak2]* należy przesunąć o podaną liczbę jednostek w lewo, w przeciwnym wypadku – w prawo. Poniżej znajduje się definicja *kerningu* dla pary liter **L** i **T**:

KPX L T -83

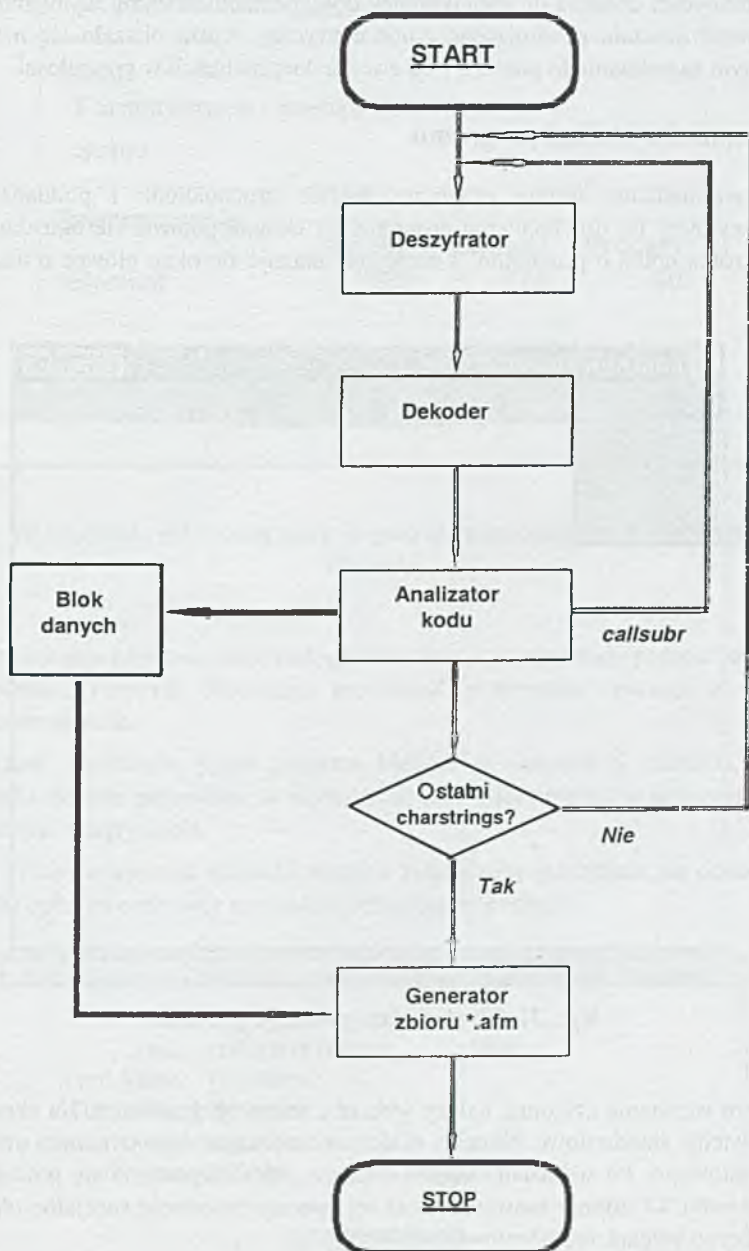
Ostatnia sekcja **D** zawiera informacje na temat liter złożonych z więcej niż jednego elementu. Przykładem takich znaków są wszystkie polskie litery generowane przez program. Linia opisująca pojedynczy znak złożony prezentuje się następująco:

CC [znak] #n ; PCC [komponent1] #dx1 #dy1 ; ... ; PCC [komponentN] #dxN #dyN ;

Grupa pierwsza rozpoczynająca się symbolem *CC* (*Composite Character*) określa nazwę znaku i precyzuje z ilu składników jest on zbudowany. Dalej następuje *#N* grup definiujących poszczególne składniki. Rozpoczynają się one symbolami *PCC* (*Part of Composite Character*) po czym następuje nazwa znaku *[komponentN]* i przesunięcie o *#dxN* i *#dyN* jednostek względem pozycji wejściowej. A oto przykład opisujący znak **Ó** składający się z litery **O** i znaku akcentu:

CC Oacute 2 ; PCC O 0 0 ; PCC acute 129 205 ;

Stosując się do wyżej opisanych reguł można opracować algorytm generujący zbiór *.afm. Po przemyśleniu i przeanalizowaniu zagadnienia został opracowany algorytm o następującej strukturze:



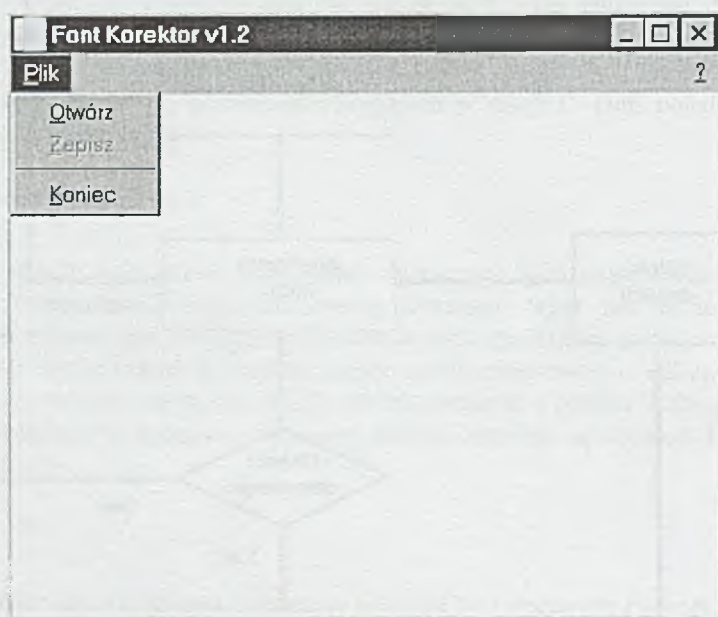
Rys. 20. Algorytm generujący zbiór *.afm

1.5.5. Moduł generacji zbioru metrycznego PFM

Ostatnim elementem programu jest moduł generacji zbioru *.pfm. Mimo usilnych starań nie udało się jednak zdobyć informacji na temat formatu tego zbioru. Wobec braku możliwości dotarcia do jakiegokolwiek dokumentacji na wyżej wymieniony temat, zrealizowanie modułu generującego zbiór metryczny *.pfm okazało się niemożliwe. Tym samym zagadnienie to pozostaje do ewentualnej realizacji w przyszłości.

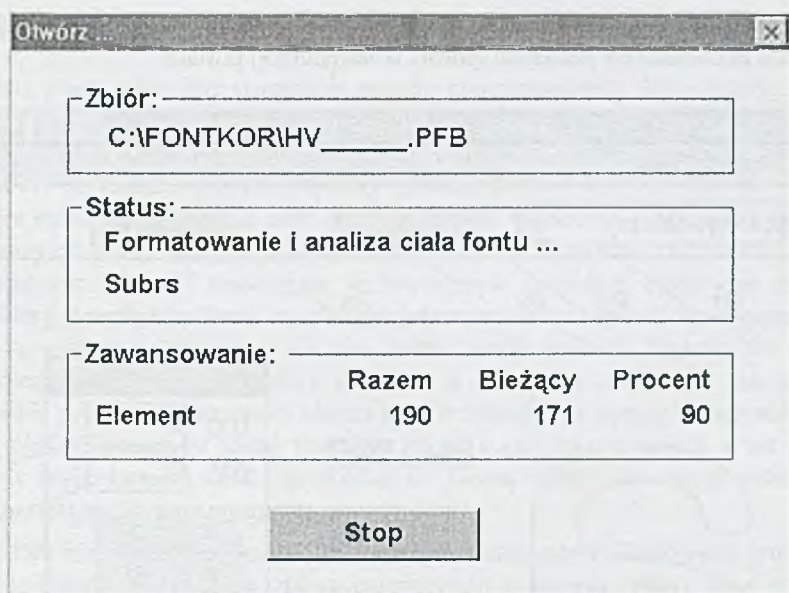
2. Opis sposobu obsługi programu

Eksperymentalna wersja programu została uruchomiona i poddana próbom eksploatacyjnym. Po uruchomieniu programu na ekranie pojawia się okienko informacyjne z krótką notką o programie, a następnie ukazuje się okno główne o następującej postaci:



Rys. 21. Wygląd okna głównego programu

Celem wczytania czcionki, należy wybrać z menu opcję Otwórz. Na ekranie pojawia się wtedy standardowe okienko dialogowe z zapytaniem o nazwę otwieranego zbioru fontowego. Po wskazaniu odpowiedniego pliku rozpoczyna się proces wczytania czcionki. O stopniu zaawansowania tej operacji informuje specjalne okno dialogowe, którego wygląd zaprezentowano poniżej:

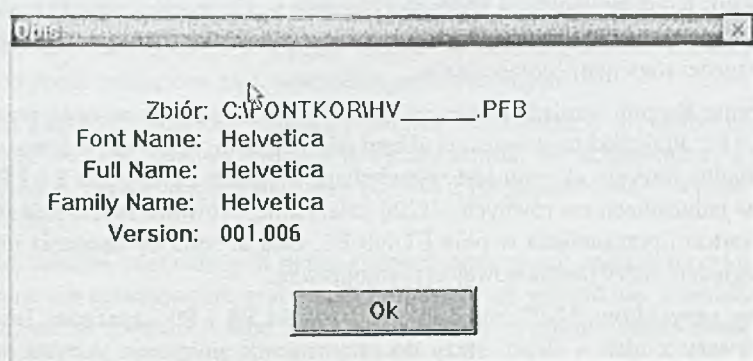


Rys. 22. Wygląd okna informującego o stopniu zaawansowania operacji wczytywania czcionki

Informuje ono użytkownika o rodzaju aktualnie wykonywanej operacji i o stanie jej zaawansowania. Przycisk Stop daje możliwość przerwania operacji w dowolnie wybranym momencie.

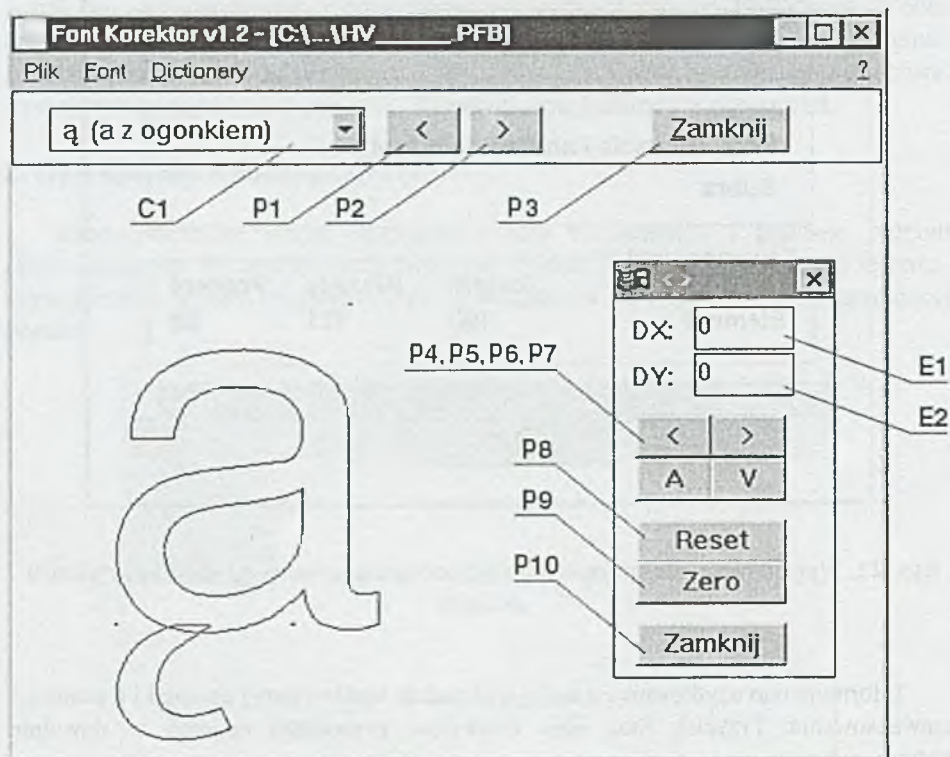
W razie napotkania przez program błędów w konstrukcji czcionki, operacji wczytywania zostaje przerwana, a użytkownik jest informowany o przyczynie przerwania operacji wczytywania.

Gdy proces otwierania czcionki zostanie zakończony pomyślnie, na ekranie ukazuje się okienko opisujące atrybuty czcionki o następującej postaci:



Rys. 23. Wygląd okienka opisującego atrybuty czcionki

Po tych czynnościach wstępnych można przystąpić do edycji polskich znaków. Do tej operacji przeznaczony jest ekran główny o następującej postaci:



Rys. 24. Wygląd okna głównego przeznaczonego do edycji polskich znaków

W górnej części ekranu znajduje się *combobox* oznaczony symbolem C1. Przy jego pomocy można dokonać wyboru jednej spośród czternastu polskich liter, którą chcemy poddać edycji. Kontury wybranej litery są rysowane w oknie głównym. Przyciski P1 i P2, znajdujące się obok *combobox'a* C1, służą do wyboru kolejnej i poprzedniej litery, bez konieczności rozwijania *combobox'a*.

Położenie akcentu względem litery podstawowej ustala się za pomocą przycisków P4, P5, P6 i P7. Przyciski te przesuwają akcent odpowiednio: w prawo, w lewo, w górę i w dół. Aktualna pozycja akcentu jest wyświetlana w polach edycyjnych E1 i E2 i jest wyrażona w jednostkach *em* równych 1/1200 cala. Istnieje również możliwość ręcznego wpisania wartości przesunięcia w pola E1 lub E2. Oba te sposoby ustalania położenia akcentu względem litery podstawowej są równoprawne.

Poniżej przycisków P4-P7 znajdują się przyciski P8 i P9 oznaczone jako *Reset* i *Zero*. Pierwszy z nich – *Reset*, służy do przywracania położenia akcentu do takiej pozycji w jakiej znajdował się przed przystąpieniem do edycji. Przycisk P9 o nazwie *Zero* ustawia zerowe przesunięcie akcentu, tj. zeruje wartości pól E1 i E2.

3. Wnioski

Celem pracy miało być stworzenie pakietu oprogramowania, które dałoby osobie pracującej z fontami Adobe Type 1 możliwość wygenerowania spolonizowanej wersji czcionki. Oprogramowanie to miało być proste w obsłudze, a zarazem funkcjonalne, tak by osoba nie posiadająca głębokiej wiedzy na temat budowy czcionek formatu Type 1 mogła bez trudu posługiwać się nim. Analizując efekty końcowe, można stwierdzić, że podstawowy cel pracy został osiągnięty. Powstało oprogramowanie, za pomocą którego można tworzyć czcionki zawierające w swym ciele wszystkie, poprawnie złożone polskie litery. Interfejs aplikacji został zaprojektowany w taki sposób by zapewnić jak największą wygodę posługiwania się nim. Jednocześnie aplikację wyposażono w wyłącznie niezbędne elementy i funkcje, tak by była ona prosta w obsłudze, zachowując jednocześnie pełną funkcjonalność. Można więc stwierdzić, że wymóg funkcjonalności i prostoty obsługi programu został spełniony, choć ocena taka, padająca z ust autora pracy jest stwierdzeniem dość subiektywnym. Ocenę należałoby raczej pozostawić osobie korzystającej z tego programu.

Niestety nie udało się zrealizować jednego z elementów składowych programu, odpowiedzialnego za generację zbioru metrycznego o nazwie *Printer Font Metrics*. Spowodowane to było brakiem jakiegokolwiek dokumentacji na ten temat. Problem ten pozostaje więc otwarty, do ewentualnego rozwiązania w przyszłości. W chwili obecnej do generacji brakującego zbioru można wykorzystać inne, istniejące aplikacje takie jak np. Fontographer.

W czasie projektowania aplikacji starano się uzyskać jak największy stopień automatyzacji procesu obróbki zbiorów fontowych, tak by spełnić wymagania mówiące o tym, że posługiwanie się aplikacją nie powinno wymagać zbyt dużego wysiłku i nakładu pracy. Większość czynności wykonywana jest automatycznie bez udziału użytkownika. Jedynymi operacjami, które musi wykonać osoba pracująca z aplikacją jest wczytanie czcionki, ustalenie położenia akcentów względem liter podstawowych, a następnie zapis nowej czcionki z polskimi literami.

Do zagadnień, którymi ewentualnie możnaby się zająć w przyszłości, w sytuacji gdyby program okazał się dla wielu osób związanych z poligrafią użyteczny, należy zaliczyć problem *hinting'u*. Być może okaże się, że generacja polskich liter pociągnie za sobą konieczność wprowadzania nowych parametrów *hinting'u*. To, czy zaistnieje taka konieczność czy nie, stwierdzić będzie można dopiero po użyciu czcionek wygenerowanych przez program w zastosowaniach profesjonalnych.

Wśród elementów, o które można by program wzbogacić w przyszłości, mogłaby znaleźć się pomoc kontekstowa on-line, tak by użytkownik nie zaznajomiony z tematem czcionek Adobe Type 1, mający pierwszy kontakt z programem, mógł szybko nauczyć się jego obsługi.

Wśród założeń postawionych przed rozpoczęciem pracy znalazł się wymóg, aby projektowane oprogramowanie pracowało poprawnie ze wszystkimi czcionkami, niezależnie od wersji formatu Adobe Type 1 i producenta. Przeprowadzone zostały liczne testy na tę okoliczność, dając pozytywne rezultaty.

Jeśli chodzi o wymagania sprzętowe, to zostały one w pełni spełnione. Program Font Korektor pracuje poprawnie na komputerach wyposażonych w procesory

począwszy od 486. Wymogi odnośnie pamięci są dość niewielkie. Do pracy aplikacja wymaga około 300 kB pamięci RAM i kilkudziesięciu kilobajtów pamięci przestrzeni dyskowej.

Literatura

- [1] Czajkowski R., Nowakowski W.: Fonty Adobe Type 1. Techniki Komputerowe Biuletyn Informacyjny, Nr 1, 1997.
- [2] Czajkowski R., Nowakowski W., Pańnikowski M.: Analiza możliwości automatycznej generacji polskich znaków w fontach postscriptowych. Opis programu – część pierwsza. Techniki Komputerowe Biuletyn Informacyjny, Nr 1, 1998.
- [3] Czajkowski R., Nowakowski W., Pańnikowski M.: Analiza możliwości automatycznej generacji polskich znaków w fontach postscriptowych. Opis programu – część druga. Techniki Komputerowe Biuletyn Informacyjny, Nr 1, 1999.
- [4] Brotz D., Paxton B., Walden J.: Adobe Type 1 Font Format / Adobe Systems Incorporated. Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1993.
- [5] Pańnikowski M.: Analiza możliwości automatycznej, programowej syntezy znaków polskich w fontach formatu Adobe Type 1. Praca dyplomowa na Wydz. Elektroniki Politechniki Warszawskiej wykonana pod kier. dr inż. Wojciecha Nowakowskiego, Warszawa.

MAREK KACPRZAK

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

Metody symulacji komputerowej i ich przydatność w inżynierii procesów biznesowych

Methods of computer simulation and their usefulness in business process engineering

Streszczenie

W pracy przedstawiono skrótowo cele i metodykę symulacji komputerowej. Omówiono możliwości zastosowania symulacji komputerowej w inżynierii procesów biznesowych. Przedstawiono celowość zastosowania symulacji na przykładzie dwóch rzeczywistych obiektów – przedsiębiorstwa produkcyjnego i handlowego.

Abstract

In this paper goals and methodology of computer simulation are briefly presented. Possibilities of applications of computer simulation in business process engineering are discussed. Two real-life applications of computer simulation for manufacture company and trade enterprise are also presented.

1. Cel i metodyka symulacji komputerowej

Symulacja komputerowa jest metodą komputerowego wspomagania projektowania nowych systemów lub analizy istniejących systemów w celu ich zmiany.

Isota symulacji komputerowej polega na badaniu nie samych systemów, ale ich modeli – sformalizowanych opisów systemów. Modele systemów zależą od czasu i parametrów (decyzji) określanych przez projektanta. Na podstawie modelu systemu tworzony jest komputerowy model systemu. Komputerowy model systemu, dla ustalonych przez projektanta parametrów (decyzji), jest wykonywany przez komputer. Wykonywanie modelu polega (najczęściej) na imitacji upływu czasu. Jedno wykonanie modelu, zwane przebiegiem symulacji, jest jednym eksperymentem na modelu.

Metodyka symulacji komputerowej [1] obejmuje następujące etapy:

1. Określenie zadania symulacji dla rozważanego systemu
2. Sformułowanie modelu systemu

3. Utworzenie komputerowego modelu systemu i programu symulacyjnego
4. Sprawdzenie poprawności modelu
5. Zaplanowanie eksperymentów symulacyjnych
6. Wykonanie eksperymentów symulacyjnych
7. Analiza i interpretacja uzyskanych wyników

1.1. Określenie zadania symulacji dla rozważanego systemu

Ustalane są kryteria oceny systemu, potrzebne wskaźniki wydajności, dostępne parametry modelu (decyzje). Na przykład, projektowany jest sposób obsługi klientów w supermarkecie. Kryterium oceny jest wynik finansowy supermarketu, zależny od liczby kas – parametru modelu. Wskaźnikami wydajności są: stopień wykorzystania każdej z kas, średni czas obsługi klienta w supermarkecie (suma czasu oczekiwania w kolejce do kasy plus czas obsługi w samej kasie). Intensywność napływu klientów, zmienna w różnych porach dnia i w różnych dniach tygodnia, jest ustalana na podstawie obserwacji istniejących podobnych supermarketów. Należy podać optymalną liczbę kas przy założeniu, że średni czas obsługi klienta nie może przekroczyć 15 minut. Wynik finansowy musi uwzględniać wynagrodzenie dla zatrudnionych kasjerów.

1.2. Sformułowanie modelu systemu

Model jest sformalizowanym opisem projektowanego lub istniejącego systemu. Zdecydowana większość modeli zależy od czasu. Rzeczywisty czas astronomiczny jest reprezentowany w modelach przez zmienną – czas symulacyjny. Modele są opisywane przy użyciu pojęć zbioru stanów i zbioru zdarzeń. Upływ czasu symulacyjnego powoduje powstawanie zdarzeń, zdarzenia powodują zmianę stanu. W niniejszej pracy omawiane są wyłącznie modele dyskretne, tzn. takie, w których zmiany stanów następują skokowo w ustalonych punktach czasu symulacyjnego.

Cechą charakterystyczną modelowania jest możliwość tworzenia różnych modeli tego samego systemu, zależnie od punktów widzenia. Model powinien reprezentować system na odpowiednim poziomie abstrakcji – jest konieczne przyjęcie właściwej dokładności i szczegółowości, zgodnie z przyjętymi kryteriami oceny. Wymaga to odpowiedniej dekompozycji problemu, a także właściwej agregacji pewnych atrybutów systemu.

Istnieje wiele formalnych metod opisu modeli dyskretnych [2]:

- opisy sieciowe (sieci kolejkowe, sieci automatów, sieci Petri z czasem itp.),
- specjalne języki symulacyjne.

1.3. Utworzenie komputerowego modelu systemu i programu symulacyjnego

Sam model zawiera jedynie usystematyzowany, zazwyczaj uproszczony opis systemu, dostosowany do określonego zadania symulacji. Dla celów symulacji, na podstawie modelu musi być utworzony komputerowy model systemu, umożliwiający utworzenie programu symulacyjnego. Współczesne narzędzia modelowania umożliwiają dokonanie przejścia od modelu systemu do programu symulacyjnego w sposób całkowicie automatyczny.

1.4. Sprawdzenie poprawności modelu

Przed rozpoczęciem właściwych eksperymentów symulacyjnych jest konieczne wykonanie szeregu eksperymentów próbnych w celu oceny wiarygodności utworzonego modelu. Wyniki uzyskiwane z eksperymentów próbnych są porównywane z wynikami obserwacji istniejącego systemu lub z oczekiwanymi wynikami dla projektowanego systemu. Jeżeli nie uzyskuje się zgodności zadawalającej, oznacza to, że konieczna jest zmiana modelu. W przeciwnym przypadku model uznaje się za wiarygodny i możliwe jest prowadzenie właściwych eksperymentów symulacyjnych.

1.5. Zaplanowanie eksperymentów symulacyjnych

W tym etapie ustala się sposób prowadzenia symulacji i warunki zakończenia pojedynczego przebiegu symulacji. Wyniki symulacji mają w zdecydowanej większości przypadków charakter losowy, dlatego dla estymacji standardowych statystyk (takich jak: wartość minimalna i maksymalna, średnia, wariancja), wyliczanych dla określonych wskaźników wydajności, konieczne jest uzyskanie dostatecznej liczby wyników. Można to osiągnąć wielokrotnie powtarzając eksperymenty dla danego zestawu parametrów (decyzji) lub prowadząc jeden eksperyment odpowiednio długo. Warunkami zakończenia przebiegu symulacji są zazwyczaj: rzeczywisty czas trwania przebiegu, uzyskanie określonej wartości czasu symulacyjnego, spełnienie określonego warunku logicznego (np. uzyskanie przez pewną zmienną w modelu określonej wartości).

1.6. Wykonanie eksperymentów symulacyjnych

Cechą charakterystyczną symulacji komputerowej jest złożoność czasowa symulacji – długie czasy trwania eksperymentów symulacyjnych. Wynika to z różnych powodów:

- wielowymiarowej przestrzeni parametrów (decyzji) modelu i wynikającego z tego obszernego programu badań,
- potrzeby wyliczania w trakcie symulacji wielu statystyk (nie tylko standardowych, ale także np. momentów wyższych rzędów) dla wielu wskaźników wydajności (takich jak przepustowość, czasy obsługi itp.),
- potrzeby określania przedziałów ufności dla średniej, wariancji i momentów wyższych rzędów, co wymaga wielokrotnego powtórzenia eksperymentów symulacyjnych lub prowadzenia jednego eksperymentu dostatecznie długo.

Jednakże, dzięki wykonywaniu symulacji sekwencyjnej na szybkich komputerach jednoprocessorowych lub symulacji równoległej (tzn. wykonywaniu jednego przebiegu symulacyjnego równocześnie na więcej niż jednym procesorze) na komputerze wieloprocessorowym lub zbiorze połączonych komputerów, rozwiązywanie praktycznych problemów przy użyciu metod symulacji komputerowej jest wykonalne.

1.7. Analiza i interpretacja uzyskanych wyników

W przypadku losowego charakteru modelu symulacyjnego jest konieczna obróbka statystyczna uzyskanych wyników i przedstawienie ich w zwartej, czytelnej postaci.

1.8. Symulacja komputerowa – zalety i wady

Podstawowe zalety metod symulacji komputerowej to:

- możliwość predykcji wydajności systemu przed jego zmianą lub implementacją,
- możliwość wielokrotnego powtarzania eksperymentów bez ingerencji w modelowany system,
- nieinwazyjny i nieniszczący charakter eksperymentów,
- nieograniczony obszar zastosowań: informatyka, ekonomia, medycyna, ochrona środowiska itd.

Wadą jest możliwość uzyskiwania wyników symulacji trudnych do zinterpretowania. Sytuacja taka występuje w przypadku niestabilnej pracy modelowanego systemu lub w przypadku przyjęcia złego modelu stabilnego systemu.

2. Stosowanie symulacji komputerowej w inżynierii procesów biznesowych

Typowa metodyka inżynierii procesów biznesowych, rozumiana zarówno jako ulepszanie istniejących (ang. business process improvement), jak i opracowanie nowych (ang. business process reengineering) składa się z następujących etapów ([3], [4]):

1. Określenie wizji i misji
2. Opis istniejącego systemu
- 3. Stwierdzenie co można udoskonalić w istniejącym systemie**
- 4. Projekt systemu docelowego**
5. Analiza wykonalności systemu docelowego i oczekiwanych efektów
6. Plan przejścia do systemu docelowego
7. Implementacja systemu docelowego
8. Eksploatacja systemu docelowego

Obszarem stosowania symulacji jako narzędzia komputerowo wspomaganey inżynierii procesów biznesowych są etapy 3 i 4. W odniesieniu do opisanej w rozdz. 1 metodyki symulacji komputerowej, należy zwrócić uwagę na specyfikę inżynierii procesów biznesowych:

1. W praktyce, określenie zadania symulacji dla rozważanego systemu – przedsiębiorstwa jest związane z koniecznością samodzielnej analizy stanu faktycznego, ze względu na brak niezbędnej dokumentacji przedsiębiorstwa (nawet jeżeli dokumentacja jest, to nie jest dostatecznie sformalizowana do celów symulacji).
2. Typowe przyjmowane kryteria to: koszt, wydajność i jakość (towarów i usług).
3. Do formułowania modeli przedsiębiorstw są stosowane wyspecjalizowane metody modelowania, np. grupy IDEF [5]. Oferowane obecnie na rynku narzędzia programowe do tworzenia tych modeli gwarantują automatyczne wygenerowanie programu symulacyjnego bezpośrednio z utworzonego modelu.

3. Przykłady zastosowań symulacji komputerowej

Podane niżej przykłady dotyczą rzeczywistych (a nie hipotetycznych), badanych przez autora przedsiębiorstw: mleczarni i przedstawicielstwa zagranicznej firmy farmaceutycznej. Są one typowymi przykładami dla szerokiej klasy przedsiębiorstw produkcyjnych i handlowych.

3.1. Przedsiębiorstwo produkcyjne

Działalność obecna [6]:

- zakup surowców i półproduktów,
- produkcja produktów, półproduktów, odpadów i ścieków,
- sprzedaż produktów i półproduktów.

Specyfika procesu produkcyjnego:

- duże zużycie limitowanej wody w procesie produkcyjnym,
- losowe wielkości dostaw surowców i półproduktów,
- losowe zamówienia,
- zasoby dzielone, niezbędne w procesie produkcyjnym (zbiorniki, linie do pakowania),
- losowe zakłócenia procesu produkcyjnego (awarie urządzeń technologicznych),
- konieczność operatywnego sterowania produkcją (decyzje podejmowane na bieżąco przez operatora).

Program naprawczy:

Cel zmian: poprawa konkurencyjności, utrzymanie pozycji na rynku, co w przypadku tego przedsiębiorstwa oznacza wymóg szybkiej realizacji zamówień.

Cel użycia symulacji:

- wykrycie wąskich gardeł w przedsiębiorstwie, których likwidacja może wymagać zmian organizacyjnych, a nawet nowych inwestycji (zakup urządzeń produkcyjnych, zbiorników, linii do pakowania),
- analiza opłacalności budowy własnej oczyszczalni ścieków,
- badania czasu realizacji nietypowego zamówienia,
- badanie wpływu zastosowania sterowania operatywnego produkcją, wspomaganego na bieżąco komputerem (praca w trybie on-line), na wyniki ekonomiczne.

3.2. Przedsiębiorstwo handlowe

Działalność obecna:

- obsługa dostawców i odbiorców (zamówienia, fakturowanie, płatności, kredytowanie, monitowanie, karne odsetki, reklama),
- planowanie (budżet, sprzedaż, koszty, zatrudnienie, inwestycje, cash-flow, dostawy z uwzględnieniem stanów magazynowych),
- kontroling (analizy finansowe dla Zarządu i właściciela),
- dokumentacja kadrowa,
- rozliczenia z fiskusem (deklaracje podatkowe, transakcje między podmiotami powiązаныmi finansowo),

- sprawozdawczość dla GUS i NBP,
- deklaracje dla ZUS,
- składki na PFRON.

Program naprawczy:

Cel zmian: poprawa organizacji przedsiębiorstwa, maksymalizacja efektów ekonomicznych.

Cel użycia symulacji:

Rozpatrywane były dwa problemy.

- I. Sprawdzenie, czy istniejąca organizacja przedsiębiorstwa zapewnia terminową realizację zadań:
 - czy są komórki organizacyjne, stanowiące wąskie gardła przedsiębiorstwa,
 - ile powinno być komórek organizacyjnych,
 - czy jest możliwość optymalizacji zatrudnienia,
 - czy nie ma dublowania pewnych prac,
 - czy jest jednoznaczna odpowiedzialność za realizację zadań,
 - jak działa przedsiębiorstwo w warunkach zakłóceń (dostaw, płatności itp.).
- II. Sprawdzenie, czy w przedsiębiorstwie jest właściwy obieg dokumentów:
 - czy są dokumenty zbędne,
 - czy nie brak jakiś typów dokumentów,
 - jaki jest czas załatwienia określonej sprawy,
 - czy są zbędne cykle w obiegu dokumentów, tzn. czy dokument wraca do pracownika kilkakrotnie,
 - czy jest przestrzegana fundamentalna zasada jednorazowego wprowadzania danych do systemu komputerowego,
 - poprawność procedur normalnej realizacji zadań dla każdego pracownika,
 - poprawność procedur awaryjnych dla każdego pracownika (np. działania w przypadku awarii systemu komputerowego).

Warto zwrócić uwagę na fakt, że utworzenie dla tego przedsiębiorstwa dwóch odrębnych modeli, zamiast jednego wspólnego jest korzystniejsze (choć oczywiście droższe – w sensie nakładu pracy i kosztów) – uzyskuje się klarowny, wolny od redundancji opis konkretnego problemu, a uproszczony model jest łatwiejszy do symulacji i analizy.

4. Wnioski

Metody symulacji komputerowej są zasadniczym narzędziem do projektowania i analizy procesów biznesowych. Należy podkreślić, że:

1. W rzeczywistych problemach projektowania i analizy przedsiębiorstw uzyskanie rozwiązania problemu drogą analityczną jest możliwe jedynie w bardzo rzadkich przypadkach, a symulacja jest jedynym narzędziem prowadzenia eksperymentów w celu uzyskania rozwiązania. Stosowanie symulacji jest zawsze konieczne, jeżeli w modelu przedsiębiorstwa występuje co najmniej jedna z następujących cech:

zmienne losowe, zasoby dzielone, zakłócenia, podejmowanie decyzji przez operatora przedsiębiorstwa.

2. Symulacja komputerowa nie określa jednak samego algorytmu rozwiązywania problemu – przyjęcie wartości parametrów lub odpowiednich decyzji zależy wyłącznie od projektanta, który w zdecydowanej większości przypadków musi polegać na doświadczeniu i intuicji. Wobec stosowania metod heurystycznych uzyskuje się rozwiązania quasi-optymalne, a nie optymalne.
3. Zarówno modelowanie, jak i symulacja są narzędziami wymagającymi specjalistycznych kwalifikacji.

Literatura

- [1] Ed. Lavenberg S. S.: Computer performance modelling handbook, Academic Press, 1983.
- [2] Tyszer J.: Symulacja cyfrowa, WNT, Warszawa, 1990.
- [3] Kaczmarczyk A.: Metody IDEF i ich zastosowanie do BPR, Biuletyn Informatyczny Techniki Komputerowe, 2/2000, s. 39-47.
- [4] Mayer R. J., deWitte P. S.: Delivering results: evolving BPR from art to engineering. Artykuł przygotowany do publikacji w książce Business Process Reengineering wydawnictwa Kluwer.
- [5] Kaczmarczyk A.: IDEF – metody modelowania i projektowania do komputerowo wspomaganey inżynierii biznesu. Część I – Informatyka 11/98, Część II – Informatyka 12/98.
- [6] Kacprzak M., Kaczmarczyk A., Slesicki M. 1998. Application of Estelle language for SME modelling: a model for examination of a creamery tolerance from temporal water shortage. *Proceedings of 9th IFAC Symp. On Information Control in Manufacturing INCOM'98* (Nancy – Metz, Francja, June 24–26), vol.III, s. 293-297.

ANDRZEJ KACZMARCZYK

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

Metody *IDEF* i ich zastosowanie do *BPR* *IDEF* methods and their application to the *BPR*

Streszczenie

Metody *IDEF* zostały opracowane w USA do modelowania systemów produkcyjnych, a znajdują zastosowanie i do innych klas systemów, w tym systemów biznesowych; stanowią skuteczny środek w *BPR*. W referacie omówiono metody: modelowania funkcji *IDEFO* oraz modelowania informacji *IDEFI*, obydwie już ujęte w federalnych normach na przetwarzanie informacji *FIPS*.

Abstract

IDEF methods have been elaborated in the U.S.A. for modeling of manufacturing processes, however they are used for modeling of systems of other classes also, business systems as well; they are an effective tool of *BPR*. In the paper the following methods are presented function modeling *IDEFO* and information modeling *IDEFI*, both already captured in federal standards of information processing *FIPS*.

1. Wprowadzenie

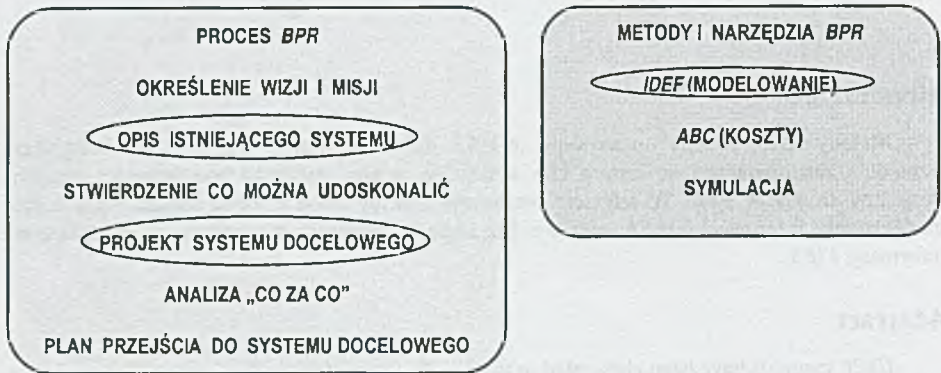
W syntetycznym skrócie technikę *BPR* (*Business Process Reengineering*) można określić jako połączenie *podejścia procesowego* i technologii informacji i komunikacji (TIK) spełniającej rolę *technologii umożliwiającej*.

Podejście procesowe polega na tym, że biznes traktowany jest jako proces: zestaw działań przekształcających „wejścia” w użyteczne dla odbiorcy „wyjścia”.

Użycie TIK jako technologii umożliwiającej rozumie się jako użycie TIK do robienia w biznesie rzeczy, które bez niej są w ogóle niemożliwe. A mianowicie:

- informacja może pojawiać się naraz w tak wielu miejscach, jak to jest potrzebne (dzięki wspólnym bazom danych);
- „generalści” mogą wykonywać pracę specjalistów (systemy eksperckie);
- podejmowanie decyzji może być częścią pracy na każdym stanowisku (narzędzia wspomagające podejmowanie decyzji);

- plany mogą być uaktualniane na bieżąco (wysoko-wydajne procedury obliczeniowe);
- personel terenowy może wysyłać i odbierać informacje w miejscu, w którym się znajduje (łąčność bezprzewodowa i przenośne komputery);
- przedmioty mogą same informować gdzie są (automatyczna identyfikacja i śledzenie);
- możliwe jest osiągnięcie korzyści jednocześnie z centralizacji i decentralizacji (wykorzystanie sieci telekomunikacyjnych);
- możliwe jest użycie interakcyjnych wideodysków zamiast kontaktów osobistych z klientem w każdej sprawie.



Rys. 1. Proces *BPR* i środki do jego przeprowadzenia

Na rys. 1 przedstawiono zadania w procesie *BPR* i środki służące do ich realizacji, przy czym wśród środków wskazano *IDEF* jako metody modelowania; uczyniono tak dlatego, że właśnie metody *IDEF* są przedmiotem tego referatu, ale możliwe jest używanie w *BPR* również i innych środków. Metody *IDEF* znajdują zastosowanie przede wszystkim do:

- opisu istniejącego systemu;
- projektowania systemu docelowego.

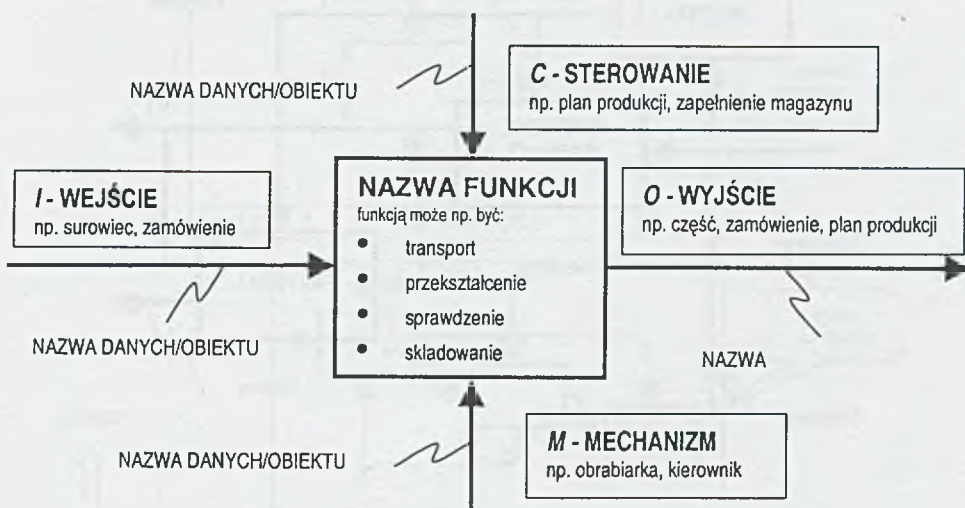
Środki komputerowego wspomagania przedsięwzięć *BPR* oferowane są na ogół w postaci pakietów zawierających podręczniki stosowanych metod, szkolenie i programy narzędziowe. Pakiet taki obejmuje całość środków pokazanych na rys. 1; znajduje zastosowanie w wykonaniu praktycznie wszystkich zadań przedsięwzięcia *BPR* ujętych na rysunku.

2. *IDEF0* – metoda modelowania funkcji

Metody *IDEF* są metodami graficznymi, tzn. modele są przedstawiane w postaci schematów.

IDEFO służy do tworzenia „modelu funkcji” będącego reprezentacją funkcji, działań, czynności zachodzących w przedsiębiorstwie lub innym systemie.

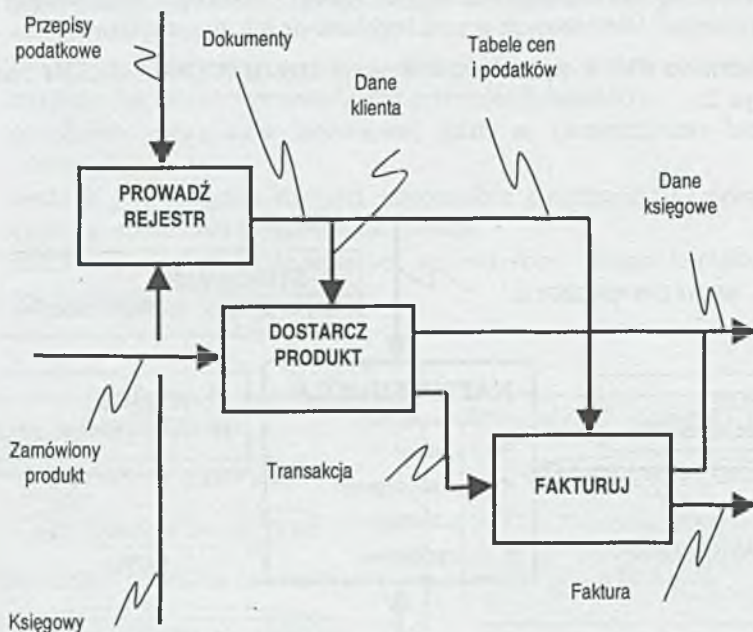
W metodzie IDEF0 wykorzystano koncept „kostki ICOM” (ICOM box), pokazany na rys. 2.



Rys. 2. Kostka ICOM

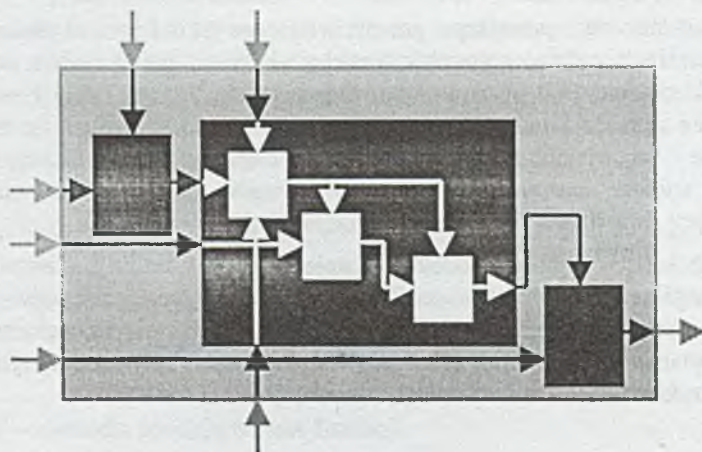
Kostka składa się z prostokąta, przedstawiającego jakiś fragment funkcjonowania systemu i czterech rodzajów strzałek, przedstawiających powiązania z otoczeniem. Nazwa kostki pochodzi od początkowych liter angielskich nazw oddziaływań obrazowanych przez strzałki: *Input*, *Control*, *Output*, *Mechanism*. Obiekty – materialne lub informacyjne – pojawiające się na *wejściu* są transformowane przez funkcję do postaci stanowiącej *wyjście*; *sterowanie* określa wykonywaną funkcję lub na nią wpływa; *mechanizm* jest narzędziem lub zasobem potrzebnym do jej wykonania.

Schemat będący zapisem modelu powstaje przez połączenie kostek wyobrażających poszczególne funkcje, w ten sposób, że wyjścia jednych kostek są wejściami lub sterowaniami innych, następuje też łączenie się lub rozdzielanie strzałek, np. w przypadku korzystania z tych samych zasobów do wykonywania różnych funkcji. Na rys. 3 pokazano przykład schematu zaczerpnięty z normy [4].



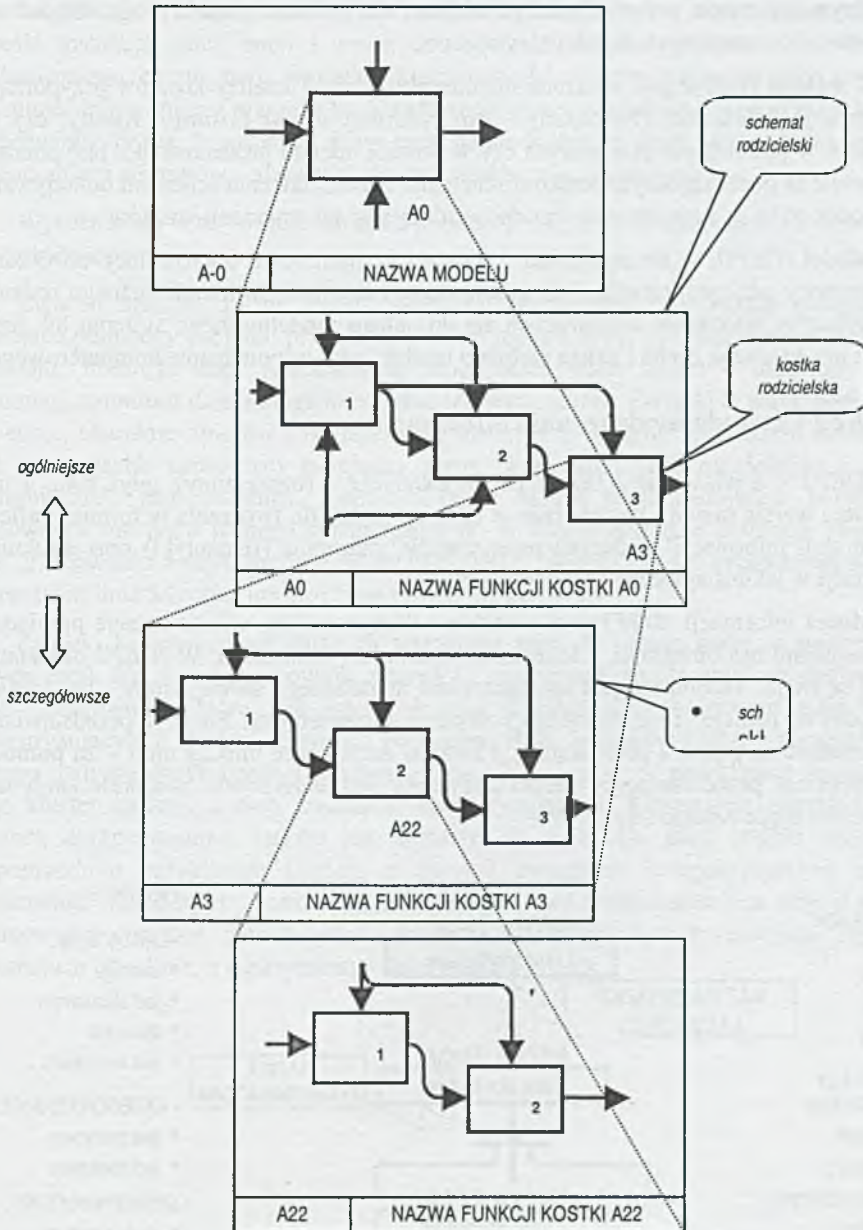
Rys. 3. Przykład schematu *IDEFO* – realizacja zamówienia (wg [4])

Istotną zasadą przyjętą w metodzie *IDEFO* jest zasada *dekompozycji*. Każda kostka ICOM może być zdekomponowana, rozwinięta w ten sposób, że zostaje przedstawiona jej wewnętrzna struktura, tak jak to pokazano na rys. 4.



Rys. 4. Zasada dekompozycji

Model *IDEF0* jest hierarchiczny, tzn. składa się z serii schematów ułożonych w ten sposób, że schematy poziomu wyższego (rodzicielskie) są rozwijane – dekomponowane – na coraz bardziej szczegółowe schematy niższych poziomów (potomne). Strukturę modelu przedstawiono na rys. 5.



Rys. 5. Struktura modelu *IDEF0*

Schematem najwyższego poziomu dla każdego modelu jest *schemat kontekstowy* (oznaczony *A-0*), zawierający tylko jedną kostkę (*A0*), stanowiącą najbardziej syntetyczne ujęcie funkcji całego modelowanego systemu. Kostka ta jest następnie dekomponowana zgodnie z zasadami przedstawionymi na rysunku; schematy potomne mogą być utworzone dla każdej kostki – stającej się wtedy kostką rodzicielską. Liczba kostek na jednym schemacie nie powinna być większa niż 6. Duże modele mogą składać się z sub-modeli utworzonych w powyższy sposób.

Z metodą *IDEFO* jest związana metoda określania i analizy kosztów przyporządkowująca je działaniom (funkcjom) – *ABC (Activity-Based Costing)*. Koszty, czy to wyrażone w jednostkach pieniężnych czy w postaci udziału procentowego, przyporządkowywane są poszczególnym kostkom schematu. Przekształcenia schematu dokonywane w procesie *BPR* są stowarzyszone z odpowiadającymi im zmianami kosztów.

Model *IDEFO*, w szczególności w postaci komputerowo operowalnej, stworzony przy pomocy jakiegoś narzędzia programowego, umożliwia tworzenie różnego rodzaju list, wykazów, wykresów odnoszących się do całego modelowanego systemu lub jego części; jest to ważna cecha i zaleta zarówno modelu jak i wspomaganie komputerowego.

3. *IDEF1* - metoda modelowania informacji

IDEF1 – a właściwie *IDEF1X* (*X* od *eXtended* – rozszerzony), gdyż taka, a nie pierwotna wersja metody jest obecnie w użyciu – służy do tworzenia w formie graficznej „modeli informacji” będących reprezentacją znaczenia (semantyki) oraz struktury informacji w jakimś systemie czy środowisku.

Model informacji *IDEF1* jest *modelem relacyjnym*, tzn. ujmuje relacje pomiędzy interesującymi nas obiektami, z którymi związana jest informacja. W *IDEF1* obiektami takimi są *encje*. Termin ten jest spolszczeniem angielskiego słowa „entity”, oznaczającego „coś co istnieje” i nie mającego polskiego odpowiednika. Encje są przedstawiane na schemacie za pomocą prostokątów, a *związki* zachodzące między nimi – za pomocą linii. Schemat przedstawiający model nazywany jest *diagramem związków encji* lub *schematem encja-relacja* (Rys. 6).

PRZYKŁADY

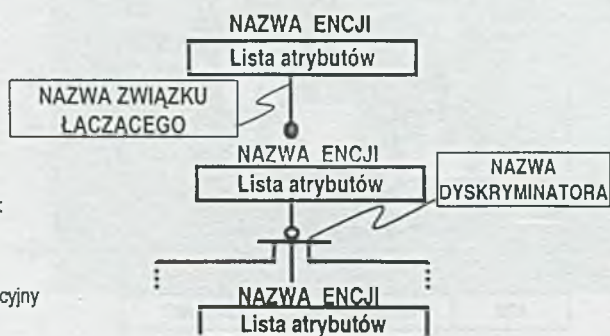
ENCJI:

- część
- dział
- klient

PRZYKŁADY

ATRYBUTÓW:

- materiał
- lokalizacja
- nr identyfikacyjny
- nazwisko



ZWIĄZKI:

- ŁĄCZĄCE, np.:

- składa
- jest składowany
- zamawia
- jest zamawiany

- KATEGORYZUJĄCE:

- jest podtypem
- jest nadtypem

DYSKRYMINATOR:

- atrybut nadtypu

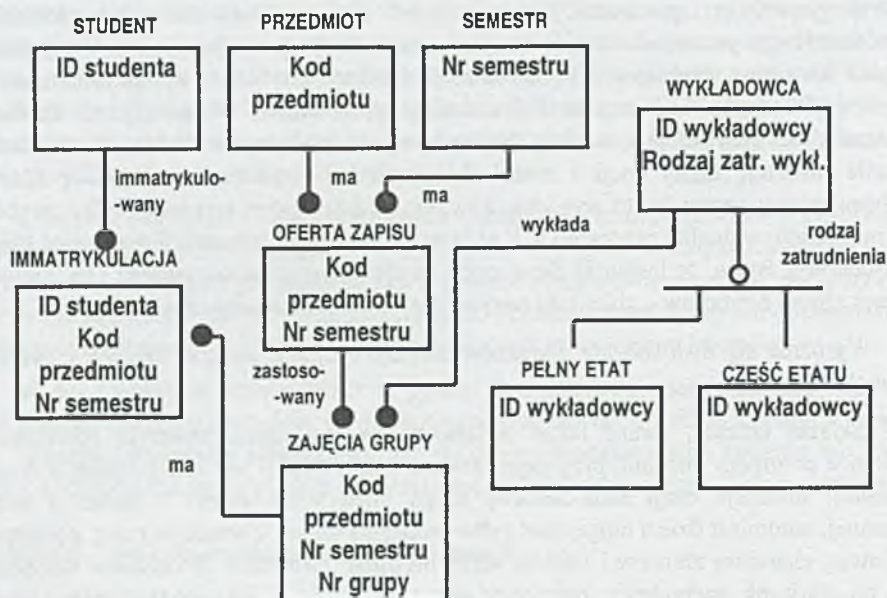
Rys. 6. Diagram związków encji

Encje to reprezentacje, w postaci zbiorów, przedmiotów rzeczywistych czy abstrakcyjnych, np.: „pracownik”, „zamówienie”. Elementy tych zbiorów – określone zamówienie czy pracownik – są *instancjami* encji. Encja jest scharakteryzowana przez *atrybut* lub grupę atrybutów. Atrybut to zbiór instancji atrybutu – np. atrybutem encji „pracownik” może być numer indywidualny pracownika, a instancjami atrybutu poszczególne, indywidualne numery. Najmniejszy zbiór atrybutów, który jednoznacznie określa instancję danej encji i został do tej funkcji wybrany, nosi nazwę *klucza podstawowego*; często, bo to wygodne, kluczem podstawowym jest jeden tylko atrybut, np. numer indywidualny pracownika. Każda encja w modelu informacji musi mieć klucz podstawowy. Bywa, że instancję danej encji da się zidentyfikować jeszcze i za pomocą innego zbioru atrybutów – zbiór taki nazywa się *kluczem alternatywnym*.

Wyróżnia się dwa rodzaje związków między encjami: *związki łączące* i *związki kategoryzujące*.

Związki łączące, zwane także związkami *rodzice-dzieci*, wyrażają powiązania istniejące pomiędzy encjami, przy czym zawsze jedna z encji jest *rodzicielska*, a druga *potomna*; instancje encji rodzicielskiej mogą mieć wiele dzieci – instancji encji potomnej, natomiast dzieci mogą mieć tylko jednego rodzica. Związki te mają, podobnie jak encje, charakter zbiorów i istnieją także instancje związków. Związkiem łączącym jest np. związek zachodzący pomiędzy encją „kupujący”, jako rodzicielską i encją „zamówienie”, jako potomną. Kupujący może złożyć wiele zamówień, natomiast zamówienie ma tylko jednego składającego je. W tym przykładzie instancją związku jest: „J. Kowalski złożył zamówienie nr 174/2000”. Związki łączące przedstawia się na schematach linią z kropką umieszczoną przy encji potomnej.

Związki kategoryzujące służą do wyrażenia tego, że pewne encje są *podtypami* innych encji. Np. encje „pracownik etatowy” i „pracownik na zlecenie” są podtypami encji „pracownik”, która jest ich *nadtypem*. Nadtyp jest skojarzony związkiem kategoryzującym z każdym ze swoich podtypów; zbiór związków kategoryzujących dla danego nadtypu nazywany jest *klasterem* typu. W powyższym przykładzie występuje więc klaster złożony z dwu związków kategoryzujących. Klaster jest określony za pomocą *dyskryminatora*, którym jest wybrany do tego celu jakiś atrybut nadtypu. W poprzednim przykładzie klasteru z dwoma związkami kategoryzującymi encję „pracownik” na podtypy: „etatowi” i „na zlecenie” dyskryminatorem jest atrybut encji „pracownik”: „rodzaj zatrudnienia”. Związki kategoryzujące przedstawia się na schematach jako linie, a dyskryminator jako kółko z kreską.



Rys. 7. Przykład schematu IDEF1X – informacja na wydziale uczelni (wg [5])

Na rys.7 pokazano przykład schematu zaczerpnięty z normy [5]. Pełne modele informacji dla złożonych systemów mogą być komponowane ze schematów częściowych zwanych widokami (polami widzenia).

Modele IDEF1 stanowią punkt wyjścia do projektowania baz danych dla modelowanego systemu, a w przypadku modeli komputerowo operowalnych tworzonych przy pomocy narzędzi programowych często możliwe jest zautomatyzowane tworzenie bazy na podstawie modelu.

4. Geneza metod IDEF, stan obecny i perspektywa

Podstawą metodologiczną IDEF jest *analiza strukturalna* wprowadzona na przełomie lat 60/70 do tworzenia oprogramowania i projektowania systemów komputerowych. Z czasem zaczęto stosować te same metody do analizy i opisu i innych systemów, nie tylko informatycznych – rozwinęła się tzw. technika SADT (*Structured Analysis and Design Technique*).

W pięcioleciu 1978-83 lotnictwo Stanów Zjednoczonych realizowało w amerykańskim przemyśle lotniczym program komputeryzacji produkcji samolotów p.n. ICAM (*Integrated Computer Aided Manufacturing*). Ujawniła się przy tym potrzeba stworzenia skutecznej techniki opisu i analizowania sytuacji oraz porozumiewania się ze sobą ludzi o różnych specjalnościach zaangażowanych w programie. Użyto techniki SADT, a konkretne metody modelowania nazwano IDEF od słów ICAM Definition.

Następcą programu ICAM został w latach 80/90 program IICE (*Information Integration for Concurrent Engineering*), w którym nadal stosowano i rozbudowywano metody IDEF; skrót IDEF jest dziś rozszyfrowywany jako *Integration DEFINition*.

W roku 1993 dla metod IDEF0 i IDEF1X ustanowiono w Stanach Zjednoczonych normy [4] i [5] należące do zbioru Federalnych Standardów Przetwarzania Informacji FIPS – co oznacza, że muszą być z tymi normami zgodne opracowania wykonywane dla agencji rządowych.

Powstają następne metody IDEF. Najbardziej zaawansowana, choć jeszcze nie znormalizowana, jest IDEF3 – metoda opisu procesów. IDEF3 ma pokazywać jak działa modelowany system, przez przedstawienie czasowych i logicznych związków między poszczególnymi działaniami oraz ról jakie spełniają obiekty w nich uczestniczące. Głównym konceptem metody jest *scenariusz* rozumiany jako zbiór sytuacji opisujących typową klasę problemów – np. załatwienie reklamacji klienta. Środkami prezentacji są *schematy procesowe* ukazujące związki czasowe i logiczne występujące w procesie oraz *schematy obiektowe*, pokazujące zmiany stanu uczestniczących w scenariuszu obiektów.

Literatura

- [1] Hammer M., Champy J.: *Reengineering the Corporation. A Manifesto for Business Revolution*. Nicholas Breal Publishing Ltd., London 1995.
- [2] Kaczmarczyk A.: *IDEF - metody modelowania i projektowania do komputerowo wspomaganey inżynierii biznesu*. Część I – *Informatyka 11/98*, Część II – *Informatyka 12/98*.
- [3] Yourdon E.: *Współczesna analiza strukturalna*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1996.
- [4] Federal Information Processing Standards Publication 183 *Integration Definition for Function Modeling (IDEF0)*. 1993.
- [5] Federal Information Processing Standards Publication 184 *Integration Definition for Information Modeling (IDEF1X)*, 1993.

ARTUR JAKUB JERZAK

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

Problem efektywności tworzenia oprogramowania w technikach internetowych i propozycja jego rozwiązania

The problem of creating Internet applications efficiently and the proposition of it solution

Streszczenie

Artykuł omawia trudności napotymane przy tworzeniu oprogramowania w technologii WWW realizującego dostęp do baz danych MS SQL Server. Jako jedno z rozwiązań proponuje zestaw narzędzi programistycznych stworzonych w IMM.

Abstract

The article describes some kind of difficulties encountered while creating Internet applications accessing MS SQL Server databases. A set of programming tools developed by IMM is proposed as a solution.

Wstęp

Współczesne witryny WWW korzystają na ogół z baz danych, często bardzo dużych. Konieczność zapewnienia odpowiedniej wydajności wymusza zastosowanie zaawansowanych rozwiązań informatycznych, a w szczególności nowoczesnych systemów zarządzania bazami danych. Jedną z możliwości jest użycie produktów firmy Microsoft, a mianowicie bazy danych MS SQL Server oraz serwera WWW IIS. W przypadku niezbyt dużych obciążeń można skorzystać z bezpłatnej wersji motoru bazy danych MS SQL Server 7.0 o nazwie MSDE 1.0 (Microsoft Data Engine). MSDE jest w pełni profesjonalną bazą danych. W szczególności:

- jest zgodny ze standardem SQL 92,
- posiada mechanizm logowania transakcji,
- gwarantuje integralność transakcji (także po awarii systemu),
- ma wbudowane mechanizmy odporności na błędy (fault tolerance),
- implementuje procedury wbudowane i wyzwalacze (pozwalają zwiększyć wydajność),
- posiada zaawansowany optymalizator zapytań,

- zawiera mechanizmy replikacji,
- pozwala na efektywny dostęp wielu użytkownikom jednocześnie,
- wspiera architekturę wieloprocessorową SMP.

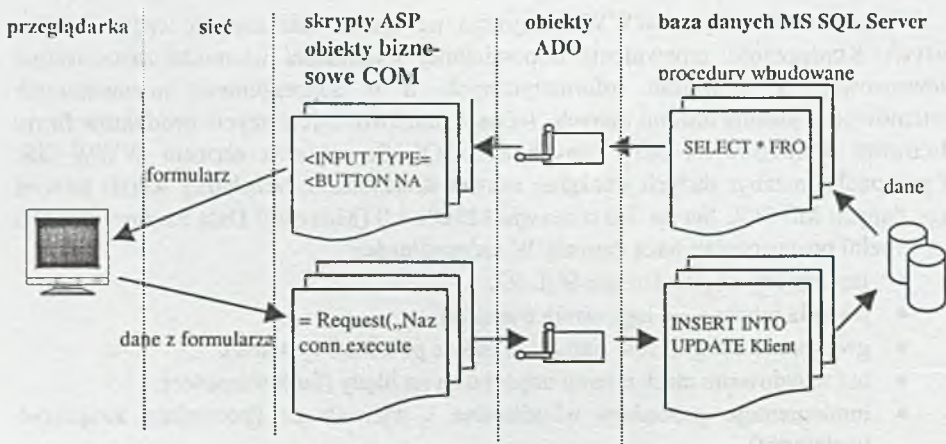
Mieliśmy możliwość praktycznego sprawdzenia tej technologii przy realizacji serwisów WWW. Doświadczenia te z jednej strony potwierdziły zalety MSDE, z drugiej zaś ujawniły niedogodności przy tworzeniu oprogramowania ograniczające wydajność pracy programistów.

Konieczność zwiększenia efektywności programistów zaangażowanych w tworzenie profesjonalnych, złożonych systemów w technikach WWW doprowadziła do powstania koncepcji opracowania narzędzi ułatwiających i przyspieszających pracę programisty. Bez nich, w sytuacji utrzymującego się nadal braku na rynku przyjaznych, zautomatyzowanych środowisk rozwojowych dla technologii WWW, nie byłoby możliwe kontynuowanie prac nad systemami WWW współpracującymi z MS SQL w tempie narzucanym przez rynek i zleceniodawców.

Dalsza część artykułu opisuje dokładniej trudności napotymane przy tworzeniu oprogramowania w technologii WWW realizującego dostęp do baz danych MS SQL Server i proponuje rozwiązanie.

Koncepcja opracowanych narzędzi wspomagających tworzenie systemów WWW

System realizujący dostęp do bazy danych przez przeglądarkę WWW w technologii proponowanej przez Microsoft składa się z wielu elementów: przeglądarki WWW, oprogramowania serwera WWW (IIS), skryptów ASP, obiektów biznesowych COM, obiektów ADO pośredniczących w dostępie do bazy danych, procedur wbudowanych (w oprogramowaniu bazy danych). W uproszczeniu z punktu widzenia programisty system wygląda następująco:



Taka wielowarstwowa struktura jest elastyczna i skalowalna. Umożliwia uniezależnienie od typu bazy danych i fizycznego rozmieszczenia tabel. Ale z drugiej strony w przypadku, kiedy baza danych jest z góry określona (MSDE) i istotna jest wydajność, programista jest zmuszony do pisania dużej ilości pośredniego kodu, który nie wnosi nic do logiki programu. Trudności wynikają także z różnic podejścia w poszczególnych warstwach i ograniczeń języków programowania. W szczególności kłopotliwa jest konwersja typów danych między bazą danych a resztą programu. Nie ma możliwości utworzenia zmiennych złożonych odpowiadających rekordowi tabeli. Można temu wprawdzie częściowo zaradzić używając obiektów typu „recordset”, ale jest to nieeleganckie i nieefektywne. Kłopoty sprawiają też ułomności języka procedur wbudowanych: brak stałych symbolicznych, niewygodne stosowanie typów definiowanych przez użytkownika, brak zmiennych złożonych. Założona niezależność od motoru bazy danych powoduje, że na programiście spoczywa konieczność zadbania o zgodność typów zmiennych dotyczących tej samej informacji (z pola tabeli) we wszystkich warstwach oprogramowania. Jest to szczególnie uciążliwe przy wprowadzaniu korekt w strukturze bazy danych, np. przy zmianie długości pól.

Dla ilustracji rozważmy bardzo prosty przykład. Niech tabela „Klient” zawiera m.in. pola:

- Id autonumer
- Nazwisko char(20)
- Imię char(15)
- Wiek int

Rozważmy teraz części programu biorące udział w rejestracji nowego klienta. Procedura wbudowana może wyglądać następująco:

```
CREATE procedure KlientInsert (
    @Nazwisko char(20),
    @Imię char(15),
    @Wiek int,
    ...
)
AS
insert into Klient(
    Nazwisko,
    Imie,
    Wiek,
    ...
)values (
    @Nazwisko,
    @Imie,
    @Wiek,
    ...
)
```

Kod procedury w komponencie COM:

```
Public Sub InsertRecord(fields, conn)
Dim cmd As New ADOB.Command

cmd.ActiveConnection = conn
cmd.CommandText = "KlientInsert"
cmd.CommandType = adCmdStoredProc

cmd.Parameters.Append
cmd.CreateParameter("@ID", adInteger, adParamInput, 0)
cmd.Parameters.Append
cmd.CreateParameter("@Nazwisko", adVarChar, adParamInput, 20)
```



```

cmd.Parameters.Append
cmd.CreateParameter("@Imie", adVarChar, adParamInput, 15)
cmd.Parameters.Append
cmd.CreateParameter("@Wiek", adInteger, adParamInput, 0)

...

cmd.Parameters("@Nazwisko") = Trim(fields("Nazwisko"))
cmd.Parameters("@Imie")     = Trim(fields("Imie"))
cmd.Parameters("@Wiek")     = Trim(fields("Wiek"))

...

cmd.Execute

End Sub

```

Część kodu ASP powodująca wyświetlenie pól formularza:

```

Nazwisko: <INPUT TYPE="TEXT" NAME="Nazwisko" MAXLENGTH="20" SIZE="20"><BR>
Imie: <INPUT TYPE="TEXT" NAME="Imie" MAXLENGTH="15" SIZE="15"> <BR>
Wiek: <INPUT TYPE="TEXT" NAME="Wiek"><BR>

```

Widać wyraźnie reguły tworzenia tego kodu. Można go więc w większości wygenerować automatycznie na podstawie deklaracji tabeli. Programiście zostałyby tylko zaprojektowanie wyglądu ekranu z zaznaczeniem odniesień do bazy danych. Powyższa strona ASP mogłaby wyglądać na przykład tak:

```

...
Nazwisko: $INPUT( „Klient.Nazwisko” )<BR>
Imie:     $INPUT( „Klient.Imie” )<BR>
Wiek:     $INPUT( „Klient.Wiek” )<BR>
...

```

Cała reszta może być utworzona automatycznie.

Znaczna część programów obsługujących bazy danych w technologii WWW zajmuje się takimi typowymi sprawami jak tworzenie i modyfikacja rekordów, sprawdzanie poprawności danych wpisanych przez użytkownika programu, prezentacja wyników wyszukiwania w postaci stronicowanych list z możliwością ograniczania liczby wyszukanych rekordów, tworzenie statystyk czy obsługa słowników systemu. Zastosowanie generatorów kodu może więc znacznie zwiększyć efektywność pracy programisty.

Pozostaje jeszcze wybór języka programowania, przy którego pomocy można sprawnie taki generator kodu napisać. Pożądane cechy to:

- wygoda operowania na tekstach: łączenie, wycinanie, przeszukiwanie (wyrażenia regularne), podstawianie,
- możliwość dosłownego wpisywania wieloliniowych tekstów (bez dodatkowych cudzysłówów, kropek itp.),
- wygodne tworzenie złożonych struktur danych: tablic, rekordów i ich kombinacji (np. tablic rekordów),
- wygodne operowanie na listach obiektów: łączenie, przeszukiwanie, powtarzanie zadanej operacji dla każdego elementu listy, sortowanie itp.,
- wsparcie „obiektywności”,
- możliwość dostępu do bazy danych MSDE,
- dostępność na platformie „Windows”.

Warunki te spełnia język Perl w wersji (dystrybucji) ActivePerl. W skład pakietu wchodzi, obok interpretera, bogaty zestaw popularnych modułów bibliotecznych (CPAN). W szczególności moduł „Win32” umożliwia korzystanie z obiektów COM, a więc i z ADO. Dzięki temu istnieje dostęp do bazy danych MSDE.

Jako rezultat powyższych założeń powstał „Zestaw narzędzi wspomagających tworzenie aplikacji WWW korzystających z bazy danych MS SQL Server”. Narzędzia te są obecnie wykorzystywane i nadal rozwijane w IMM.

WOJCIECH KUBERA

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

Internet w kieszeni, czyli wprowadzenie w technikę WAP

Internet in pocket or introduction to WAP technology

Streszczenie

WAP (Wireless Application Protocol – protokół aplikacji bezprzewodowych) jest standardem, który zapewne już wkrótce zaowocuje niespotykanym dotąd wzrostem liczby i jakości usług informacyjnych świadczonych przez operatorów sieci łączności bezprzewodowej.

Abstract

WAP (Wireless Application Protocol) is the standard which certainly soon will set with unknown as yet increase of number and quality of information services rendered by operators of wireless communication network.

Geneza

Idea mobilności jest jedną z najbardziej nośnych koncepcji w technice. Pierwszym spektakularnym przykładem był chyba konkurs na sposób „przewożenia czasu”, który doprowadził w XVII wieku do wynalezienia chronometru i niebywałego rozkwitu nawigacji morskiej. Wynalazek czołgu – ruchomej fortecy – miał istotny wpływ na sposób prowadzenia wojny. Przykłady można mnożyć. W miarę postępu techniki powstawało coraz więcej urządzeń przenośnych. Kiedy byłem chłopcem, fascynował mnie pomysł zbudowania przenośnego odbiornika radiowego i gdyby nie to, że tak zwane radio kryształkowe wymagało długiej, wysoko zawieszanej anteny, pewno udało by się mi go zrealizować. Dzisiaj przenośne urządzenia elektroniczne są czymś zupełnie oczywistym i powszechnie używanym. Przenośność nie ominęła oczywiście techniki komputerowej: powstały komputery walizkowe, kieszonkowe i cała gama współpracujących z nimi urządzeń. Do pełni szczęścia brakowało już tylko przenośnego, bezprzewodowego Internetu. Paradoksalnie, katastroficzne wizje społeczeństwa bez reszty pogrążonego w rzeczywistości wirtualnej, nie ruszającego się na krok od komputerowych ekranów, komunikującego się służbowo i prywatnie za pośrednictwem globalnej sieci, czerpiącego z sieci strawę duchową i rozrywkę – te wizje chyba się jednak nie spełnią. Internet stał się czymś niezbędnym, ale to nie Internet nas przywiąże do siebie – to my będziemy go wszędzie zabierać ze sobą.

Wielką szansą dla mobilnego dostępu do Internetu stał się żywiolowy rozwój telefonii komórkowej. Jej zasięg powiększa się w bardzo szybkim tempie, a dodatkowym atutem jest możliwość przesyłania danych w formie cyfrowej. Skonstruowano więc wyrafinowane urządzenia sprzęgające przenośny komputer z telefonem komórkowym. Dla wygody wykorzystano możliwość bezprzewodowego połączenia przez port pracujący na podczerwień, opracowano specjalne oprogramowanie, jednak rezultat jest daleki od oczekiwań. Telefon komórkowy nie zapewnia dostatecznej szybkości przesyłania danych cyfrowych (obecnie używane standardy umożliwiają przesyłanie danych z szybkością 9600 bodów) i jeśli przesłanie krótkiej wiadomości pocztą elektroniczną odbywa się w czasie możliwym do przyjęcia, to łączenie się z serwisami informacyjnymi WWW jest po prostu mordęgą w dodatku dosyć kosztowną. Stało się jasne, że przynajmniej dopóki nie nastąpi znaczny wzrost szybkości transmisji danych cyfrowych przez telefon komórkowy, trzeba szukać innych efektywniejszych metod korzystania z zasobów globalnej sieci.

W połowie 1997 roku sprawę wzięli w swoje ręce potentaci telefonii komórkowej. Cztery firmy: Ericsson, Nokia, Motorola i Unwired Planet zawiązały WAP Forum otwartą organizację, której celem było opracowanie standardów niezbędnych do efektywnego połączenia sieci telefonii komórkowej z Internetem. Założenia były następujące:

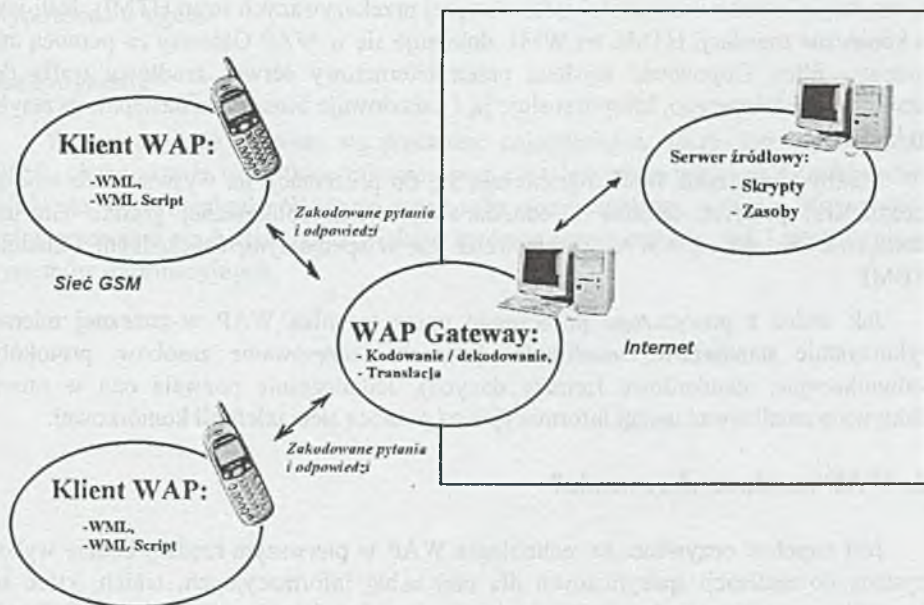
- Nowa technika powinna możliwie dużo przyjąć z Internetu, jego sprawdzonych standardów, możliwości i zasobów;
- Dostęp do serwisów informacyjnych powinien być możliwy za pośrednictwem mikroprzeglądarki, dla której platformą sprzętową będzie tylko nieznacznie zmodyfikowany telefon komórkowy lub aparat przywoławczy (pager), a więc moc obliczeniowa, którą można dysponować, będzie niewielka, pojemność pamięci mocno ograniczona, zobrazowanie danych trzeba zrealizować na monochromatycznym, bardzo małym wyświetlaczu, wprowadzanie danych musi być możliwe za pomocą klawiatury telefonu;
- Powinny być wzięte pod uwagę skromniejsze możliwości sieci bezprzewodowych w porównaniu z sieciami kablowymi: mniejsza szybkość transmisji, dłuższe opóźnienia, mniejsza stabilność połączenia;
- Wprowadzenie usług związanych z dostępem do Internetu nie tylko nie powinno kolidować z dotychczasowymi funkcjami i usługami, ale powinno je wspierać i otwierać nowe możliwości. Nowe standardy nie powinny być w jakikolwiek sposób uzależnione od stosowanego systemu telefonii komórkowej, tak aby mogły być zaakceptowane przez wszystkich operatorów – tylko takie podejście mogło zaowocować szybkim rozwojem nowych usług informacyjnych świadczonych za pośrednictwem sieci komórkowych.

Inicjatywa WAP została dobrze przyjęta w środowisku operatorów sieci bezprzewodowych i producentów telefonów komórkowych. Do WAP Forum przystąpiło już około 350 członków. Prace nad specyfikacją WAP zaczęły przynosić praktyczne rezultaty w ubiegłym roku, kiedy zaakceptowana została pierwsza wersja standardu. Jednocześnie powstało odpowiednie oprogramowanie dla Internetu i ruszyły pierwsze serwisy informacyjne, a Nokia wprowadziła do sprzedaży pierwszy telefon wspierający WAP: model 7110. Perspektywy są obiecujące. Przewiduje się, że za trzy lata będzie na

świecie działać około pół miliarda telefonów z nowymi funkcjami internetowymi. Serwisy WAP powstają również w Polsce. Według prognoz operatorów sieci komórkowych w tym roku liczba telefonów wyposażonych w mikroprzeglądarkę osiągnie u nas około 200 tysięcy.

Jak to działa?

WAP wykorzystuje klasyczny model charakterystyczny dla aplikacji WWW. Dane udostępniane przez serwis informacyjny przechowywane są w zasobach Internetu na serwerze źródłowym WWW. Serwer analizuje skierowane do niego zapytania przeglądarki (zapytania dotyczą konkretnych zasobów adresowanych za pomocą URL) i na podstawie posiadanych danych formuje odpowiedź w standardowym formacie. Ze względu na ograniczone możliwości obliczeniowe bezpośrednia współpraca mikroprzeglądarki z serwerem WWW nie jest możliwa. Dlatego też niezbędny jest serwer pośredniczący między telefonem komórkowym i serwerem WWW. Serwer ten nazywany jest WAP Gateway (brama WAP) i z punktu widzenia architektury sieci WWW może być traktowany jak serwer proxy. Strukturę sieci udostępniającej usługi WAP przedstawia rys. 1.



Rys. 1. Ogólna struktura sieci udostępniającej serwisy WAP

Również ze względu na ograniczoną moc obliczeniową i małą szybkość transmisji, nie jest możliwe interpretowanie w mikroprzeglądarce języka HTML. Zamiast niego

zdecydowano się zastosować specjalnie opracowany wariant języka XML, dostosowany do możliwości telefonów komórkowych. Językowi temu nadano nazwę WML (*Wireless Mark-up Language – język znaczników dla aplikacji bezprzewodowych*). Zdefiniowano także specjalny język skryptowy WMLScript, umożliwiający zdynamizowanie prezentowanych przez mikroprzeglądarkę informacji. Podstawową jednostką danych w WML jest talia (angielskie *deck*), składająca się z tak zwanych kart (ang. *card*) stanowiących odpowiedniki stron (ekranów). Talia jest podstawową jednostką danych przekazywaną (i ładowaną) do mikroprzeglądarki.

Celem ograniczenia wielkości strumienia danych przesyłanych przez sieć bezprzewodową kod WML trafia do mikroprzeglądarki nie w formacie tekstowym, ale w skompresowanej formie binarnej. Podobnie zapytania kierowane do Internetu za pomocą protokołu WAP kodowane są binarnie. Z kolei kod WMLScript przekazywany jest do mikroprzeglądarki w postaci przetranslowanej (także binarnej) na język zbliżony w swojej koncepcji do języków maszynowych.

Zapytania generowane przez klientów WAP (mikroprzeglądarki) trafiają do serwera pośredniczącego, który rozkodowuje je i tłumaczy na format WWW. Przetłumaczone zapytania trafiają do właściwego serwera WWW, który generuje odpowiedź. Z zasady odpowiedzi generowane przez serwisy WAP zapisane są od razu w językach WML i WMLScript. Tłumaczenie z języka HTML na WML jest możliwe, ale wiąże się ze znacznymi stratami zawartości informacyjnej przekazywanych stron HTML. Jeśli jest to konieczne translacje HTML na WML dokonuje się w WAP Gateway za pomocą tak zwanego filtra. Odpowiedź wysłana przez internetowy serwer źródłowy trafia do serwera pośredniczącego, który transluje ją i zakodowuje binarnie, a następnie przesyła do klienta.

Możliwości języka WML ograniczają się do prezentacji na wyświetlaczu mikroprzeglądarki prostych tekstów i jednobarwnej, nieskomplikowanej grafiki. Grafika bitmapowa dla aplikacji WAP, zapisywana jest w specjalnym, oszczędnym formacie WBMP.

Jak widać z powyższego pobieżnego opisu technika WAP w znacznej mierze wykorzystuje standardowe możliwości Internetu (adresowanie zasobów, protokoły komunikacyjne, standardowe formaty danych). Jednocześnie pozwala ona w miarę efektywnie zrealizować usługi informacyjne za pomocą sieci telefonii komórkowej.

Co WAP ma do zaoferowania?

Jest zupełnie oczywiste, że technologia WAP w pierwszym rzędzie będzie wykorzystana do realizacji specyficznych dla niej usług informacyjnych, takich, które są szczególnie przydatne, gdy jesteśmy poza domem. Już dziś za pomocą telefonu WAP możemy zapoznać się z najaktualniejszymi wiadomościami politycznymi i gospodarczymi, prognozą pogody, kursami walut, repertuarem kin i teatrów, rozkładami jazdy. W najbliższej przyszłości za pomocą telefonów WAP będzie możliwe przeprowadzanie operacji bankowych, rezerwacji biletów i innych usług. W zasadzie wszystkie użyteczne na co dzień informacje można udostępnić poprzez WAP. Przecież nie od rzeczy będzie się dowiedzieć czy coś, co właśnie mamy zamiar kupić nie kosztuje czasem taniej w sąsiednim supermarkecie. W razie kłopotów z dogadaniem się z przygodnie spotkaną

cudzoziemką warto sięgnąć poprzez WAP do słownika, by móc powiedzieć jej coś miłego.

Trzy przyszłościowe kierunki zastosowań WAP wydają się szczególnie interesujące. Pierwszy z nich, to serwis umożliwiający dostęp do bazy danych, którą wypełniamy sami (my osobiście lub firma w której pracujemy) i do której tylko my mamy dostęp. Wszystkie informacje jakie mogą nam być potrzebne, gdy jesteśmy poza domem, gromadzimy na wynajętym (lub firmowym) serwerze i możemy z nich skorzystać niezależnie od tego gdzie jesteśmy. Drugi z nich wiąże się z tym, że sieć telefonii komórkowej wie gdzie się znajdujemy. W związku z tym możliwe jest generowanie treści stron WML z uwzględnieniem miejsca, w którym jesteśmy, a więc dowiedzieć się o adresie najbliższej stacji benzynowej, czy rozkładzie jazdy najbliższej stacji kolejowej. Wreszcie trzeci kierunek dotyczy wykorzystania możliwości jakie daje komunikacja głosowa, do której przecież telefon jest szczególnie predysponowany. Interakcja za pomocą głosu na pewno będzie szybsza niż za pomocą klawiszy telefonu. Najnowsze badania wykazują, że rozpoznanie głosowej odpowiedzi użytkownika w przypadku, gdy musi on wypowiedzieć z góry określone słowa, nie wymaga dużych mocy obliczeniowych. Wyobraźmy sobie na przykład, że zamawiamy bilet do teatru. Serwis realizujący tę usługę odtwarza nam zapytanie „parter czy balkon?”. Oczywiście dużo łatwiej jest rozpoznać, czy wypowiedzieliśmy jedno z tych słów, niż rozpoznać naszą wypowiedź w ogóle.

Zakończenie

W tym artykule starałem się przekazać najogólniejsze fakty dotyczące techniki WAP, chcąc przede wszystkim zainteresować czytelnika tym tematem. Według oceny wielu ekspertów, technika WAP ma przed sobą okres szybkiego rozwoju. Z pewnością zainteresowanie nią będzie coraz większe zarówno wśród autorów jak i użytkowników systemów informacyjnych.

WIESŁAW BICZ
OPTEL SP. Z O.O. WROCLAW

Metody rozpoznawania odcisków palców stosowane w systemach kontroli dostępu ze szczególnym uwzględnieniem metod ultradźwiękowych

Fingerprint recognition methods used in access control systems with special emphasis on ultrasonic methods

Streszczenie

W niniejszym artykule opisana została aktualna sytuacja na rynku urządzeń biometrycznych, ze szczególnym uwzględnieniem rozpoznawania odcisków palców. Szczególną uwagę poświęcono kamerze ultradźwiękowej, pozwalającej na obserwację linii papilarnych i rozróżnienie, czy są one częścią żywego palca. Rozważono też dalsze perspektywy rozwoju urządzeń ultradźwiękowych przeznaczonych do rozpoznawania palców.

Abstract

In this paper, the situation on the market of biometrics devices is described, with special consideration of fingerprint recognition methods.

Wprowadzenie

W ciągu ostatnich kilku lat zauważyć można powstawanie nowej branży. Ma ona nawet już swoją nazwę: „biometria”. Jej twórcy pragną skonstruować urządzenia, za pomocą których możliwe byłoby identyfikowanie ludzi na podstawie ich „biologicznych” cech, czyli takich, które każdy „nosi ze sobą”. Do dnia dzisiejszego zaproponowano, zaprojektowano lub skomercjalizowano urządzenia wykorzystujące rozpoznawanie następujących charakterystycznych elementów ciała ludzkiego lub sposobów jego działania: Wzorów linii papilarnych opuszków palców, kształtu dłoni (lub pojedynczych palców), struktur tęczówki lub siatkówki (dna oka), rysów twarzy, głosu, podpisu, dynamiki ruchów, posiadanej wiedzy i umiejętności.

Dodatkowo można by jeszcze wymienić propozycję rozpoznawania przebiegu żył na dłoni, zrealizowaną nawet – przynajmniej w formie eksperymentalnej – oraz

wykorzystania zapachu, kształtu ucha, paznokci lub kodu genetycznego, które na razie chyba nie wyszły jeszcze poza etap propozycji.

W niniejszym artykule chciałbym najpierw omówić pokrótce metody wspomniane w punktach 2-7, a potem bardziej szczegółowo rozpoznawanie linii papilarnych.

Opis przydatności i perspektyw zastosowania poszczególnych metod

Rozpoznawanie kształtu dłoni jest metodą stosunkowo prostą w użyciu, łatwą do zastosowania w niesprzyjających warunkach (np. przy brudnych rękach pracowników warsztatu). Jej wadą jest niezbyt duża rozpoznawalność. Urządzenia tego typu są sprzedawane od lat i mają swój dość ugruntowany rynek. W razie rozpowszechnienia się również niekłopotliwych w działaniu urządzeń gwarantujących większą pewność rozpoznawania można się spodziewać eliminacji tego typu urządzeń.

Rozpoznawanie struktur wewnątrz gałki ocznej jest metodą dość kłopotliwą – wymaga przyłożenia oka do okularu, co wielu ludzi robi niechętnie, bojąc się ewentualnych konsekwencji. Rozpoznawanie tęczy jest prostsze, bo możliwe z pewnego dystansu. Obie metody obarczone są głównie dwiema wadami: nie ma pewności, czy rozpoznawane struktury na pewno nie powtarzają się u różnych ludzi i nie jest możliwe zabezpieczenie się w prosty sposób przed oszustwami za pomocą atrap. Mimo to urządzenia tego typu są oferowane od kilku lat, niektórzy pokładają w nich nawet dość duże nadzieje np. ze względu na możliwość rozpoznawania tęczy z pewnej odległości.

Rozpoznawanie rysów twarzy ma – podobnie jak tęczy, jedną niewątpliwą zaletę: Jest wykonywalne zdalnie – z pewnej (nawet dużej) odległości. Daje to szansę identyfikacji ludzi na zdjęciach uzyskanych przy pomocy standardowych kamer. Wiadomo jednak na pewno, że nie daje ono 100% pewności rozpoznania. W wielu przypadkach może być jednak wygodne lub nawet konieczne (kiedy ma się tylko wizerunek twarzy do dyspozycji). Prawdopodobniejsze wydaje się rozpoznanie w przypadku ujmowania przestrzennej struktury głowy, urządzenia tego typu są jednak dość kosztowne. W chwili obecnej dostępne są zarówno proste rozwiązania w postaci programu wykorzystującego standardowe kamery, jak i też złożone systemy, używające wielu kamer, specjalnego oświetlenia i wyrafinowanych procedur.

Rozpoznawanie głosu jest jedną z takich metod, które pozostają jako jedyna możliwość w niektórych przypadkach – na przykład wtedy, kiedy chodzi o rozpoznawanie ludzi korzystających z telefonu. Z tego powodu metody te są intensywnie rozwijane i zawsze będą miały swoje miejsce na rynku. Nie należy się jednak spodziewać, że kiedykolwiek będzie możliwe uzyskanie przy ich pomocy aż takiej pewności rozpoznania, jaka osiągalna jest przy pomocy odcisków palców. W wielu przypadkach nie jest to jednak konieczne, a rozpoznawanie mówiącego może bardzo ułatwić wiele zadań.

Rozpoznawanie podpisu jest chyba najstarszą „metodą biometryczną”, którą ludzie wykorzystywali zdalnie (tzn. bez bezpośredniego kontaktu z drugim człowiekiem). Od lat wiele firm rozwija tę metodę, proponując urządzenia i programy. Jest ona interesująca szczególnie dlatego, że korzysta ze starej tradycji używania podpisów w bankowości i do zatwierdzania umów. Automatyczne rozpoznawanie podpisu pozwala

na uchwycenie nie tylko kształtu podpisu, ale także jego dynamiki, co zwiększa szansę rozpoznania. Trudno jednak powiedzieć, czy metoda ta zwycięży w porównaniu z innymi.

Rozpoznawanie dynamiki ruchów jest metodą, którą ludzie potrafią stosować dość bezbłędnie, ale jej implementacja w urządzeniach jeszcze najwyraźniej nie dojrzała do praktycznego zastosowania. Metoda może być interesująca z takich samych powodów jak rozpoznanie twarzy – możliwa jest jej implementacja jako software'u, analizującego obraz ze zwykłych kamer video.

Wykorzystanie posiadanej wiedzy i umiejętności jest jedną z najstarszych „metod biometrycznych”. Czymże bowiem innym jest stosowany powszechnie PIN, nie mówiąc już o szyfrowanych zamkach, ukrytych dźwigniach itd. Zaproponowano wiele znacznie bardziej wyrafinowanych metod tego typu, ale wygląda na to, że pozostają one w dalszym ciągu swego rodzaju kuriozum w Świecie klasycznej biometrii.

Urządzenia i metody stosowane do rozpoznawania odcisków palców

Po tym skrótowym opisie istniejących już metod chciałbym wrócić do metody najbardziej mnie interesującej, ale też i niewątpliwie dominującej na dzisiejszym rynku biometrycznym: do rozpoznawania linii papilarnych. Z metodą tą wielu fachowców wiąże największe nadzieje, i to nie przypadkiem. Niewątpliwie jest bowiem, że charakterystyczna struktura linii papilarnych jest inna u każdego człowieka, jak też to, że nie zmienia się ona w ciągu jego życia. Dotykaniem palcem powierzchni sensora jest poza tym bardzo prostą czynnością. Pragnieniem niejednego wynalazcy urządzeń biometrycznych jest więc skonstruowanie klawisza, który „wie”, kto go przyciska, i jakim palcem to robi. Użyty do otwierania drzwi klawisz taki wpuszczałby oczywiście tylko „swoich”, a z tego właśnie chce żyć cała ta branża [1 - 3].

Od wielu już lat istnieją urządzenia optyczne, pozwalające na odczytywanie struktury linii papilarnych na żywo – wprost z palca – bez wykorzystywania tuszu i papieru – sprawdzonych, ale niezbyt wygodnych w użyciu narzędzi policjantów od czasów twórców daktyloskopii: Henry'ego, Herschela, Fauldsa i Galtona [4] [5]. Metody optyczne cechuje jednak wiele wad: Bezpośredni obraz palca ma znikomy kontrast, łatwiej na nim zobaczyć brud niż linie papilarne; obraz trójwymiarowy jest trudno wykonalny i zawodzi w przypadku zniszczonych powierzchni palców [2]; wykorzystanie odbicia światła od powierzchni, do której przyłożony jest palec jest natomiast bardzo wrażliwe na tłuszcz, brud i wodę (i to zarówno na ich brak, jak też i nadmiar). Żadna metoda optyczna nie pozwala na łatwe stwierdzenie, czym naprawdę jest obserwowana struktura: czy jest to prawdziwy, żywy palec, jego atrapa, czy też może pozostawiony na powierzchni sensora ślad – tłuszczowy (klasyczny) lub odpowiednio spreparowany odcisk palca... (Opis działania typowych urządzeń tego rodzaju zawiera [6]).

Nic dziwnego więc, że pojawiły się też inne niż optyczne sposoby odczytywania struktury linii papilarnych, np. zaproponowana przez Constantine Tsikosa metoda pojemnościowa [7], rozwijana ostatnio przez kilka firm, które produkują i sprzedają realizujące tą metodę układy scalone. Mają one głównie tę zaletę, że są mniejsze i tańsze od urządzeń optycznych, nie są jednak od nich lepsze: są bardziej wrażliwe na

uszkodzenia (struktura chipu jest ukryta pod bardzo cienką warstwą ochronną) i na przepięcia. Nie dają one też lepszych obrazów. Poza ceną ich zaletą może być fakt, że nie zauważają one pozostawionych na powierzchni sensora śladów (klasycznych odcisków palca) [8].

Pojawiły się też odmiany metod pojemnościowych (np. zaproponowana przez firmę Authentec, używająca zmiennych pól elektromagnetycznych). Zaproponowane zostały też rozwiązania z optyką światłowodową, pozwalające na znaczny stopień integracji urządzenia.

Pewną ciekawostką jest metoda zaproponowana przez Thomson CSF, wykorzystująca linijkę sensorów piroelektrycznych. Chip ten wymaga przesuwania palca względem niego i wskutek tego ma dość specyficzne własności, głównie jednak wady (najważniejsze są przy tym zniekształcenia obrazu przy tym powstające). Wydaje mi się, że należy go traktować bardziej jako kuriozum niż sensowną propozycję.

Zupełnie niezależną klasę urządzeń tworzą te, które wykorzystują ultradźwięki. Ponieważ wzbudzają one moje szczególne zainteresowanie, a poza tym jestem przekonany, że oferują one niespotykane w innych przypadkach perspektywy rozwoju, to chciałbym poświęcić ich omówieniu dużą część artykułu.

Możliwości urządzeń ultradźwiękowych

Członkowie zespołu polskiej firmy Optel są prekursorami metod wykorzystujących ultradźwięki do rozpoznawania palców (pierwsza propozycja [14] pochodzi z roku 1986). Poza nimi jeszcze tylko jedna firma amerykańska (Ultrascan) zajmuje się takimi urządzeniami. W swojej najbardziej wyrafinowanej formie pozwalają one na łatwe odróżnianie prawdziwych, żywych palców od wszelkich innych rzeczy, nie są wrażliwe na brud, tłuszcz itp., nie przeszkadza im zniszczona powierzchnia palca, stwarzają poza tym dodatkowe perspektywy, niewyobrażalne dla innych metod: możliwe jest np. wykonanie urządzenia posiadającego dość dowolnej wielkości i kształtu powierzchnię reagującą na dotyk (także wielu palców), mogącego stwierdzać ich położenie, identyfikować je, rejestrować ruch. Urządzenie takie nie posiadałoby żadnych części ruchomych i oczywiście byłoby w stanie zastąpić dzisiejsze klawiatury, myszy, urządzenia do identyfikacji palców, choć na tym jego możliwości bynajmniej się nie kończą. Dla uzupełnienia obrazu warto wiedzieć, że możliwe jest też zrobienie urządzenia, które będzie małe, tanie (rodzaj chipu) i rzeczywiście da się zainstalować w przycisku. Urządzenie takie może mieć jeszcze jedną ciekawą zaletę: jest możliwa taka jego wersja, która pozwala na zdalną identyfikację ludzi (np. przez sieć), i to takich, którzy dysponują dowolnymi możliwościami technicznymi, nie istnieje bowiem możliwość jego oszukania.

Na temat sposobu działania opracowanych urządzeń firmy Optel napisane zostały artykuły [9 - 13]; zgłoszonych i udzielonych zostało też kilka patentów (m. in.: [14 - 16]; ich właścicielem, jak też i posiadaczem praw komercyjnych do urządzenia jest firma Sonident, Vaduz).

Zasada działania urządzeń ultradźwiękowych

W chwili obecnej znane są dwie zasady działania takich urządzeń:

- Wykorzystanie zogniskowanej wiązki ultradźwiękowej, skanującej mechanicznie powierzchnię kontaktu palca z ciałem stałym;
- Kamera holograficzna, obserwująca z wielu kątów rozproszenie ultradźwięków na powierzchni kontaktowej;

W obydwu przypadkach wykorzystuje się zjawisko, które można w skrócie opisać w sposób następujący:

Jeśli do powierzchni ciała stałego, do której dociera dźwięk przyłożony jest obiekt, i kontakt między nim a powierzchnią nie jest wszędzie jednakowy (idealny), lecz zawiera niejednorodności (krawędzie, punkty kontaktowe itp.), to w miejscach takich dojdzie nie tylko do opisanego klasycznymi wzorami przejścia dźwięku z jednego ośrodka do drugiego, jego odbicia oraz dyfrakcji na granicach obszarów kontaktu, lecz także do dotatkowego rozproszenia i przemiany na inne rodzaje fal. Jest ono wynikiem zmiany warunków propagacji dźwięku w pobliżu powierzchni ciała stałego, spowodowanej kontaktem z przyłożonym do niej obiektem, dlatego też nazywać je będziemy rozproszeniem kontaktowym. Jest rzeczą pewną, że wpływają na nie nie tylko same obszary styku obu ośrodków, lecz także zbliżona do nich część przyłożonego obiektu (w dalszym ciągu tego artykułu nazywana strukturą przypowierzchniową). Z tego też zapewne powodu zjawisko to zależne jest silnie od materiału, z którego wykonany jest przyłożony obiekt.

Doświadczenia pokazują, że przejście fali z jednego ośrodka do drugiego może praktycznie nie występować, lecz obserwowane jest jedynie rozproszenie i generacja innych rodzajów fal (szczególnie wyraźnie dostrzegalne jest to w przypadku fal poprzecznych). Uzasadnione wydaje się twierdzenie, że powstające w miejscach kontaktu z przyłożonym obiektem zaburzenia fali mają charakter głównie fazowy (jej czoło ulega przestrzennemu odkształceniu) i że właśnie takie zmiany frontu falowego są bezpośrednią przyczyną obserwowanego rozproszenia.

Konstrukcja kamery holograficznej, perspektywy jej rozwoju

W chwili obecnej urządzenie wykorzystuje jedną głowicę nadawczo-odbiorczą, poruszającą się w wodzie. Możliwa jest jego wersja, wykorzystująca wiele nieruchomych przetworników. Opis urządzenia zawiera [17]. Prawdziwie interesujące jest jednak urządzenie, które nazywamy „czułą szybą”, którego podstawowym elementem będzie właśnie szyba (lub inny kawałek materiału litego), którego powierzchnia służyć będzie jako miejsce do przykładania palców lub dłoni.

Łatwo jest zauważyć, że płytką taką będzie „widzieć” nie tylko opuszki palców, lecz także rozpoznawać ich położenie. Może ona więc zastąpić klawiaturę lub mysz. Będzie też oczywiście mogła rozpoznać miejsce przyłożenia np. ołówka i śledzić jego ruch. Da się więc przy jej pomocy zrealizować także funkcję tabliczki graficznej, co pozwala na wpisywanie rysunków, czy też ręcznego pisma. Jednak do identyfikacji ludzi nie wystarczyłoby to – na szczęście fale dźwiękowe mogą zarówno rozpoznawać materiały kontaktujące się z płytą, jak też wchodzić w głąb palca. Pozwala to na

stwierdzenie, czy palec jest prawdziwy i żywy – widać czy (i jak) płynie w nim krew. Nie wydaje się żeby oszuści mieli w tej sytuacji jeszcze jakieś szanse...



Rys. 1 Aktualny wygląd kamery holograficznej



Rys.2 Odcisk palca widziany przez kamerę

Literatura

- [1] Emma Newham: *The Biometrics Report*, SJB Services, ISBN 1-900-18009;
- [2] Dirk Mohrmann: *Biometrie als Quantensprung?*; W&S 7/97 (Strona 28), Hüthig Verlag, Heidelberg;
- [3] Hans Müller: *Der Körper als Passwort*; PC Magazin, Januar 1998 (S. 256), DMV Verlag, D-85622 Feldkirchen.
- [4] Francis Galton: *Finger Prints*, opublikowane 1892 przez Macmillan and Co., London i New York, reprint: Da Capo Press, New York 1965;
- [5] Henry C. Lee, R.E. Gaensslen: *Advances in Fingerprint Technology*; Elsevier 1991, ISBN 0-444-01579;
- [6] Lars H. Andersen, Peter Jürgensen: *Fingerprint Verification - for use in Identity Verification Systems*, Master Degree Work, Aalborg University 1993;
- [7] Siemens AG: EP 0 041 693 A1; 1981;
- [8] *Fingerprint-Chips*, Markt&Technik Nr. 50/97 (12.12.97, S. 46), Magna Media Verlag, D-85531 Haar;
- [9] *Der Fingertip-Sensor*; *Blick durch die Wirtschaft*, (18.12.97, S. 6), FAZ Verlag, Frankfurt am Main;
- [10] Z. Gumienny, M. Pluta, W. Bicz, D. Kosz: *Rekonstrukcja struktur dwuwymiarowych z zastosowaniem algorytmu FFT*; *Akustyka Molekularna i Kwantowa*, tom 15 (1994);
- [11] Z. Gumienny, M. Pluta, W. Bicz, D. Kosz: *Synthetic aperture acoustic microscope for evaluation of fingertip peripheral skin structure*; *SPIE Vol. 2390*;
- [12] W. Bicz, Z. Gumienny, M. Pluta: *Ultrasonic Sensor for Fingerprint Recognition*; *SPIE Vol. 2634*;
- [13] Z. Gumienny, M. Pluta, W. Bicz, D. Kosz: *Ultrasonic Setup for Fingerprint Patterns Detection and Evaluation*; *Acoustical Imaging*, Vol. 22, Plenum Press 1996;
- [14] Sonident: World Patent WO-A-8 705 790;
- [15] Sonident: US Patent Nr.: 5258922;
- [16] Sonident: US Patent Nr.: 5515298;
- [17] Wiesław Bicz, Dariusz Banasiak, Paweł Bruciak, Zbigniew Gumienny, Stanisław Gumuliński, Dariusz Kosz, Agnieszka Krysiak, Władysław Kuczyński, Mieczysław Pluta, i Grzegorz Rabiej: *Ultrasonic Camera for Finger Ridge Pattern Imaging*, *Archives of Acoustics* 23,1,139-148 (1998).

ROMUALD SYNAK

INSTYTUT MASZYN MATEMATYCZNYCH WARSZAWA

Optoelektroniczne metody i urządzenia do pomiaru zasięgu widzialności Optoelectronic methodes and instruments for visibility range measurement

Streszczenie

Przedstawiono teoretyczne podstawy pomiaru zasięgu widzialności i metody pomiaru tego parametru, a także scharakteryzowano urządzenia pomiarowe wytwarzane obecnie głównie dla potrzeb komunikacji lotniczej i drogowej.

Abstract

Theoretical fundamentals of visibility range measurements and methodes for measuring this parameter are presented in the paper. Description of instruments produced now for needs of aerial and land communication is also given.

1. Wstęp

Konieczność prowadzenia stałego monitorowania stanu przezroczystości powietrza dla potrzeb transportu lotniczego, drogowego i morskiego spowodowała rozwój automatycznych urządzeń pomiarowych pracujących w sieciach komputerowych. Miarą przezroczystości jest tzw. meteorologiczny zasięg widzialności, oznaczający odległość, przy której kontrast ciała doskonale czarnego obserwowanego na tle nieba przy horyzoncie maleje do pewnej standardowej wartości. Bezpośredni pomiar meteorologicznego zasięgu widzialności polegający na pomiarze kontrastu okazał się z przyczyn technicznych za mało dokładny. Stosowane są dlatego metody pośrednie, oparte na pomiarze tłumienia światła w atmosferze, opisanego za pomocą współczynnika ekstynkcji (pochłaniania optycznego) atmosfery. Jego znajomość umożliwiła na podstawie prawa Koschmiedera [10] obliczenie meteorologicznego zasięgu widzialności. Współczynnik ekstynkcji może być mierzony metodą transmitancyjną lub rozproszeniową. W pierwszym przypadku przyrząd (nazywany transmisometrem) mierzy transmitancję atmosfery na pewnej długości przejścia światła, a w drugim moc światła

rozproszonego przez cząstki znajdujące się w powietrzu (przyrząd służący do tego celu jest nazywany skaterometrem) [7].

Oprócz znacznego postępu w technice pomiarów meteorologicznego zasięgu widzialności, uzyskano też obiecujące rezultaty w dziedzinie pomiaru zasięgu widzialności obiektów rzeczywiście znajdujących się w terenie. Ich widzialność zależy od rodzaju obiektu, faktury powierzchni, barwy i innych cech, występowania opadów deszczu lub śniegu itp. Na zdolność zauważania przedmiotów przez człowieka wpływają nie tylko jego indywidualne cechy, wyrażane wartością progu czułości kontrastowej oka, ale także rozmiary obiektów, poziom oświetlenia ogólnego i inne [6]. Różnice między meteorologicznym zasięgiem widzialności a zasięgiem odnoszącym się do przedmiotów rzeczywistych mogą wynosić kilkadziesiąt procent. Wobec wymogu automatyzacji pomiarów metody określania zasięgu widzialności obiektów rzeczywistych oparte na wykorzystaniu wyszkolonych obserwatorów lub specjalnych przyrządów fotometrycznych okazały się obecnie nieprzydatne. Postęp przyniosło dopiero opracowanie tzw. wideografów [14] - działających na zasadzie analizy uzyskanego za pomocą kamery CCD obrazu pasa startowego lotniska lub jego otoczenia.

Rezultaty pomiarów otrzymane przy wykorzystaniu samych sensorów optoelektronicznych okazują się często niewystarczające do prawidłowego wyznaczenia zasięgu widzialności. Konieczne są w takich przypadkach pomiary dodatkowych parametrów meteorologicznych, niezbędna jest też odpowiednia komputerowa obróbka uzyskanych danych, wykorzystanie m.in. metod eksperckich [13], metod logiki rozmytej [2]. Współczesne urządzenia do pomiaru zasięgu widzialności stanowią zatem złożony system pomiarowy.

Celem artykułu jest przedstawienie problematyki związanej z budową takich systemów, a w szczególności ukazanie specyfiki wykorzystania w nich światła jako nośnika informacji o stanie przezroczystości atmosfery. Zwrócono uwagę zwłaszcza na systemy przeznaczone do pracy na potrzeby ruchu lotniczego. Oprócz pokazania osiągnięć technicznych czołowych firm zajmujących się wytwarzaniem takich urządzeń, podano również informację nt. transmisometru opracowanego w Instytucie Maszyn Matematycznych, w którym, w przeciwieństwie do zwykle stosowanego źródła światła białego (lampy halogenowej), zastosowano laser półprzewodnikowy [17].

2. Podstawy teoretyczne pomiaru widzialności

Ograniczenie widzialności w powietrzu jest wynikiem tłumienia światła przez znajdujące się w nim gazy i cząstki aerozolu. Można wyróżnić następujące zjawiska powodujące osłabienie promieniowania ([11], [18]):

- rozpraszanie molekularne,
- absorpcję molekularną,
- rozpraszanie aerozolowe.

Strata mocy wiązki światła o pewnej długości fali λ wywołana tymi zjawiskami jest proporcjonalna do mocy światła $P(\lambda)$, długości odcinka drogi dx przebytej przez światło oraz współczynnika ekstynkcji ośrodka $\alpha(\lambda, x)$:

$$dP(\lambda) = -\alpha(\lambda, x) P(\lambda) dx \quad (2.1)$$

Zależnie od zjawiska wywołującego tłumienie światła wyróżnia się współczynnik ekstynkcji dla absorpcji molekularnej, rozpraszania molekularnego oraz rozpraszania aerozolowego. Jeżeli jednocześnie oddziałują dwa lub trzy zjawiska, to całkowity współczynnik ekstynkcji jest sumą współczynników odnoszących się do tych zjawisk.

Całkując równanie (2.1) dla warunków brzegowych X_1 i X_2 oznaczających odległość od źródła światła, przy mocy początkowej $P_0(\lambda)$, otrzymuje się wyrażenie

$$P(\lambda) = P_0(\lambda) \exp\left[-\int_{X_1}^{X_2} \alpha(\lambda, x) dx\right] \quad (2.2)$$

które w przypadku, gdy ośrodek jest jednorodny, upraszcza się do postaci zwanej prawem Bouguera-Lamberta

$$P(\lambda) = P_0(\lambda) \exp[-\alpha(\lambda)D] \quad (2.3)$$

przy czym D oznacza długość drogi równą $X_2 - X_1$.

Transmitancja (transmisja) ośrodka $T(\lambda)$ wyraża stosunek mocy wiązki, która przeszła przez ośrodek do mocy początkowej, tj.

$$T(\lambda) = P(\lambda)/P_0(\lambda) = \exp[-\alpha(\lambda)D] \quad (2.4)$$

Powyższe wzory odnoszą się do światła monochromatycznego. Jeżeli wiązka światła zawiera spektrum rozciągające się od λ_1 do λ_2 , to współczynnik transmisji ośrodka jednorodnego można wyrazić w postaci wielkości średnich określonych wg następującego wzoru:

$$T_{sr} = \frac{\int_{\lambda_1}^{\lambda_2} P_0(\lambda) \exp[-\alpha(\lambda)D] d\lambda}{\int_{\lambda_1}^{\lambda_2} P_0(\lambda) d\lambda} \quad (2.5)$$

Równanie to określa transmitancję atmosfery jako funkcję strumienia świetlnego wiązki.

Zgodnie z podaną we wstępie definicją meteorologicznego zasięgu widzialności, można obliczyć go na podstawie pomiaru kontrastu ciała doskonale czarnego i nieba przy horyzoncie. Na podstawie teorii Koschmiedera [10] można sformułować następującą funkcję $C(x)$ określającą kontrast mierzony w odległości x od obiektu.

$$C(x) = \frac{\int_{\lambda_1}^{\lambda_2} R(\lambda) N(\lambda) [\exp(-a(\lambda)x)] d\lambda}{\int_{\lambda_1}^{\lambda_2} R(\lambda) N(\lambda) d\lambda} \quad (2.6)$$

gdzie przez $R(\lambda)$ i $N(\lambda)$ oznaczono odpowiednio luminancję energetyczną promieniowania tła o długości fali λ i czułość spektralną odbiornika.

Pomiar kontrastu jest obarczony na ogół dużym błędem wynikającym z trudności w dokładnym określeniu luminancji obiektu oraz tła i dlatego zasięg widzialności wyznacza się na podstawie pomiaru transmitancji ośrodka. Przedmiotem pomiaru jest strumień świetlny dochodzący do odbiornika, odniesiony do wartości strumienia w warunkach braku tłumienia światła. Funkcję transmitancji dla drogi światła x można na podstawie wzoru (2.5) wyrazić w postaci wzoru

$$T(x) = \frac{\int_{\lambda_1}^{\lambda_2} P(\lambda)S(\lambda)[\exp(-a(\lambda)x)]d\lambda}{\int_{\lambda_1}^{\lambda_2} P(\lambda)S(\lambda)d\lambda} \quad (2.7)$$

gdzie przez $P(\lambda)$ oznaczono moc promieniowania dochodzącego do detektora przy braku tłumienia światła przez ośrodek, a przez $S(\lambda)$ czułość spektralną detektora.

Podobieństwo wyrażeń (2.6) i (2.7) potwierdza możliwość wykorzystania funkcji $T(x)$ do obliczenia zasięgu widzialności, jeżeli funkcja $P(\lambda)S(\lambda)$ ma taki sam przebieg jak $R(\lambda)N(\lambda)$. Jednakże analityczne obliczenie zasięgu widzialności jako wartości x , przy której wartość $T(x)$ obniża się do poziomu progowej czułości kontrastowej oka ε nie jest wprost możliwe, gdyż funkcje występujące w podanych wzorach są zależnościami uzyskanymi na drodze doświadczalnej i z wyjątkiem funkcji $\alpha(\lambda)$, nie mają swojej reprezentacji analitycznej. Metodę obliczania zasięgu widzialności na drodze numerycznej podano w pracy [15].

W praktyce obliczenia zasięgu widzialności dokonuje się przy założeniu, że wartość współczynnika ekstynkcji jest stała w zakresie widzialnym (co jest słuszne tylko w warunkach występowania gęstej mgły) i równa się pewnej wartości α . Wzory (2.6) i (2.7) upraszczają się w tym przypadku do postaci

$$C(x) = T(x) = \exp(-\alpha x) \quad (2.8)$$

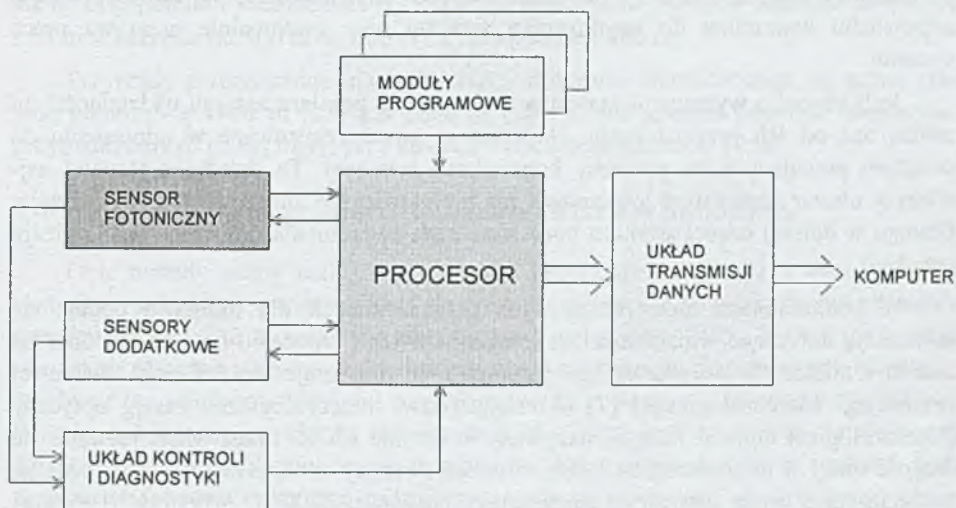
Zasięg widzialności V_m można obliczyć przez przyrównanie prawej strony równania do wartości progowej czułości kontrastowej oka ε . Otrzymuje się wówczas

$$V_m = -\ln \varepsilon / \alpha \quad (2.9)$$

Tak więc zasięg widzialności przy ustalonej wartości ε można obliczyć na podstawie wyniku pomiaru współczynnika ekstynkcji atmosfery. Jeśli chodzi o wartość współczynnika ε , to z obszernych badań dokonanych przez Blackwella [1] wynika, że ε_z wynosi $0,008 \div 0,014$. Middleton przeprowadził 1000 serii badań, na podstawie których stwierdził, że średnia wartość ε_z wynosi 0,03, przy czym duża liczba wyników wykazała wartość 0,01, a niewielka liczba wartości bardzo duże $0,1 \div 0,15$ [12]. W badaniach dotyczących atmosfery oraz pomiarach wykonywanych dla celów synoptycznych przyjmuje się wartość 0,02, natomiast przyrządy mierzące zasięg widzialności dla potrzeb komunikacji lotniczej i drogowej podają wartość zasięgu obliczoną dla $\varepsilon = 0,05$.

3. Ogólna charakterystyka systemu do pomiaru zasięgu widzialności

Schemat blokowy urządzenia do pomiaru zasięgu widzialności podano na rys. 1. Główną funkcję pomiarową pełni sensor fotoniczny, który zależnie od przyjętej metody pomiaru umożliwia określenie wartości współczynnika ekstynkcji atmosfery lub dostarcza danych fotometrycznych dotyczących obserwowanego obszaru. Wyniki pomiaru są przekazywane do procesora, którym może być typowy komputer włączony do systemu lub specjalizowany układ mikroprocesorowy.



Rys. 1. Schemat blokowy systemu pomiarowego zasięgu widzialności

Obliczenie wykonuje się zgodnie z procedurą zawartą w jednym z modułów programowych procesora. Wynik tych obliczeń jest poddawany dalszej obróbce określonej w kolejnych modułach. Przede wszystkim dokonuje się podstawowego zabiegu, jakim jest eliminacja wyników przypadkowych, obliczanie wielkości średnich itp. [7]. Główna korekta jest przeprowadzana w celu kompensacji błędów wynikających z wpływu czynników zewnętrznych. Metoda postępowania zależy od typu sensora fotonicznego. W przypadku np. urządzenia działającego na zasadzie pomiaru mocy światła rozproszonego dokonuje się analizy wyników pomiaru wybranych parametrów meteorologicznych uzyskanych za pomocą innych sensorów. Ma ona na celu określenie rodzaju mgły, która spowodowała obniżenie się zasięgu widzialności i wyznaczenie wielkości związanego z nią współczynnika korekcyjnego podającego relację między wartością zmierzonej mocy a całkowitą mocą światła rozproszonego. Jeżeli pomiar odbywa się za pomocą metody transmitancyjnej oblicza się poprawkę uwzględniającą zanieczyszczenie powierzchni optycznych. Podstawą do tego mogą być wyniki pomiaru zanieczyszczeń uzyskane za pomocą odpowiedniego sensora lub też na podstawie analizy statystycznej wyników pomiaru widzialności uzyskanych w dłuższym okresie czasu. Wyniki końcowe są przekazywane do komputera systemu nadrzędnego.

Prawidłowa praca systemu jest kontrolowana za pomocą procedur sprawdzających funkcjonowanie poszczególnych sensorów. Podstawą oceny są wartości pewnych parametrów uzyskane za pomocą czujników i układów tworzących zespół regulacji i diagnostyki. Mierzone są takie wielkości, jak temperatura panująca w wybranych miejscach sensora, poziom napięć zasilających, poziom niektórych napięć lub prądów w układach elektronicznych sensora, moc promieniowania lasera itp. Dane te są wykorzystywane do automatycznej regulacji niektórych z wymienionych wielkości. W przypadku, gdy zmiana mierzonych parametrów przekracza założony zakres, odpowiedni moduł programowy dokonuje analizy przyczyn występowania niekorzystnej zmiany, wysyła odpowiedni komunikat do użytkownika systemu oraz ewentualnie przerywa pracę systemu.

Jeśli chodzi o wymagania stawiane systemom do pomiaru zasięgu widzialności, to zależą one od ich przeznaczenia. Najlepiej są one sprecyzowane w odniesieniu do urządzeń pracujących na potrzeby komunikacji lotniczej. Ta dziedzina stanowi największy obszar zastosowań systemów i ma też największe znaczenie dla ich rozwoju. Dlatego w dalszej części artykułu poruszone będą zagadnienia dotyczące tego rodzaju urządzeń.

W komunikatach meteorologicznych przeznaczonych dla lotnictwa podaje się informację dotyczącą widzialności na lotnisku i w jego otoczeniu oraz widzialność na pasach lotniska. Co do pierwszego parametru, to przyjmuje się definicję Światowej Organizacji Meteorologicznej [7] określającą tzw. meteorologiczny zasięg optyczny (Meteorological Optical Range), nazywany w skrócie MOR. Przez MOR rozumie się długość drogi w atmosferze, na której strumień świetlny skolimowanej wiązki światła, pochodzącej z lampy żarowej o temperaturze barwowej 2700 K, maleje do 0,05 wartości początkowej, przy czym strumień jest mierzony za pomocą miernika o charakterystyce spektralnej zgodnej z funkcją fotopową podaną przez Międzynarodową Komisję Oświetlenia (CIE). Mimo, że definicja wyraźnie określa metodę pomiaru i jego warunki, w praktyce nazwa MOR jest używana również w przypadku innych warunków pomiaru, a nawet innych metod pomiaru (np. rozproszeniowej). Błąd pomiaru zasięgu widzialności powinien mieścić się w następujących granicach [19]: ± 50 m w zakresie do 600 m, $\pm 10\%$ w zakresie 600÷1500 m i $\pm 20\%$ powyżej 1500 m.

W komunikatach podaje się wartość liczbową zasięgu jedynie do 10000 m. Jeśli chodzi o wymaganą wartość dolną zakresu pomiarowego, to zależy ona od kategorii lotniska. Transmisometry przeznaczone dla lotnisk cywilnych najwyższej kategorii (IIIB) powinny mierzyć zasięg widzialności już od 10 m.

Drugi mierzony na lotniskach parametr – zasięg widzialności na pasie (Runway Visual Range, w skrócie RVR) definiuje się jako odległość, przy której pilot samolotu znajdującego się na środku pasa startowego może widzieć znaki lub światła określające granice pasa lub jego linię środkową. Zależnie od tego, czy warunki ogólnego oświetlenia umożliwiają zauważenie znaków granicznych, czy też konieczne jest włączenie światel pozycyjnych, zasięg RVR określa się na podstawie pomiaru MOR lub prawa Allarda [4], które dotyczy widzialności obiektów punktowych źródeł światła. Zgodnie z tym prawem źródło światła o światłości I w odległości D_z wywołuje wrażenie wzrokowe o natężeniu E równym

$$E = I \frac{\exp(-\alpha D_z)}{D_z^2} \quad (3.1)$$

Utrata widzialności następuje, gdy przy pewnej odległości E osiąga wartość równą progowej czułości świetlnej oka E_{pr} . Wielkość ta zależy bardzo od warunków oświetlenia ogólnego (tła) oraz od wielkości iloczynu kontrastu progowego oraz kąta, pod którym oko widzi źródło. Aby więc móc obliczyć wielkość RVR konieczne jest dokonywanie pomiaru zarówno luminancji tła, jak i współczynnika ekstynkcji. Zaleca się by błąd pomiaru widzialności RVR był nie większy niż ± 10 m w zakresie do 400 m, ± 25 m w zakresie od 400 m do 800 m i $\pm 10\%$ powyżej 800 m.

Przyrządy przeznaczone dla komunikacji drogowej charakteryzują się górną granicą pomiarową 1000 m [23] lub 2000 m [26]. Dolna granica powinna obejmować przypadki bardzo gęstej mgły, przy których widzialność spada do 10 m.

4. Systemy oparte na pomiarze tłumienia światła w atmosferze

Obie metody oceny zasięgu widzialności polegające na pomiarze współczynnika ekstynkcji atmosfery – transmitancyjna i rozproszeniowa – polegają na dokonywaniu pomiaru w pewnym określonym miejscu na powierzchni Ziemi i ekstrapolowaniu wyniku na otaczający obszar. Takie podejście jest możliwe do przyjęcia przy założeniu, że obszar ten jest jednorodny pod względem składu powietrza i warunków oświetlenia. Ponadto zakłada się zwykle, że współczynnik ekstynkcji jest wielkością niezależną od długości fali. Mimo błędów oceny widzialności, które mogą powstawać wskutek tych uproszczeń, obecnie tylko tego rodzaju przyrządy są stosowane, gdyż zapewniają one pomiar całodobowy, odbywający się nieprzerwanie i automatycznie.

4.1. Transmisometry

Zasada pomiaru. Przyrząd do pomiaru widzialności metodą pomiaru transmitancji – transmisometr (w Polsce stosowana jest również nazwa transmisjometr) – mierzy strumień świetlny wiązki światła przechodzącej przez atmosferę na odcinku od nadajnika do odbiornika. Długość tego odcinka nazywa się długością lub odległością bazową (wzgl. krócej bazą) i oznacza literą B . Jeżeli powietrze jest bardzo przezroczyste, światło nie ulega znaczącemu tłumieniu i można przyjąć, że mierzony strumień jest wtedy taki sam, jak na wyjściu nadajnika (F_o). Transmitancję atmosfery na odległości bazowej $T(B)$ można zatem obliczyć przez podzielenie aktualnie mierzonego strumienia świetlnego (F) przez uprzednio zmierzony strumień F_o . Pozwala to z kolei na obliczenie współczynnika ekstynkcji powietrza na podstawie prawa Bouguera-Lamberta. Dokonuje się tego zwykle przy założeniu, że współczynnik ekstynkcji nie zależy od długości fali w zakresie widzialnym i jest równy pewnej stałej wartości α . Dla takiego przypadku na podstawie równania (2.7) otrzymuje się następujące wyrażenie na transmitancję:

$$T(B) = F/F_o = \exp(-\alpha B) \quad (4.1)$$

z którego można obliczyć, że

$$\alpha = -\frac{\ln T(B)}{B} = -\frac{\ln(F/F_0)}{B} \quad (4.2)$$

Meteorologiczny zasięg optyczny MOR oblicza się jako dystans V_M , przy którym strumień świetlny maleje do poziomu 0,05 wartości początkowej. Korzystając znowu z prawa Bouguera-Lamberta można obliczyć, że

$$V_M = -\frac{\ln 0,05}{\alpha} = \frac{2,9957}{\alpha} \cong \frac{3}{\alpha} \quad (4.3)$$

oraz

$$V_M = -\frac{3B}{\ln T(B)} = -\frac{3B}{\ln F/F_0} \quad (4.4)$$

Podstawowe cechy metody transmitancyjnej. Z podanych wzorów wynika, że zasięg widzialności jest odwrotnie proporcjonalny do logarytmu naturalnego transmitancji. Skutkiem tego w warunkach dużej przezroczystości, gdy transmitancja zbliża się do 1, małe błędy pomiaru transmitancji powodują powstanie dużego błędu oszacowania widzialności. Ogranicza to od góry możliwy do uzyskania zakres pomiarowy widzialności. Zakres ten jest także ograniczony od dołu. Jeżeli transmitancja zmniejsza się poniżej 0,05, na jej dokładność pomiaru coraz większy wpływ mają przesunięcia poziomu stałego mierzonego sygnału i inne błędy powstające w układzie pomiarowym. W literaturze stwierdza się, że możliwy jest pomiar transmitancji z błędem ok. $\pm 1\%$ [7]. Przy założeniu, że dopuszczalny błąd oceny widzialności jest $\pm 10\%$, mierzona transmitancja powinna zawierać się w granicach od 0,03 do 0,9. Wielkość obliczanego zasięgu widzialności zależy, jak wynika ze wzoru (4.4), także od odległości bazowej.

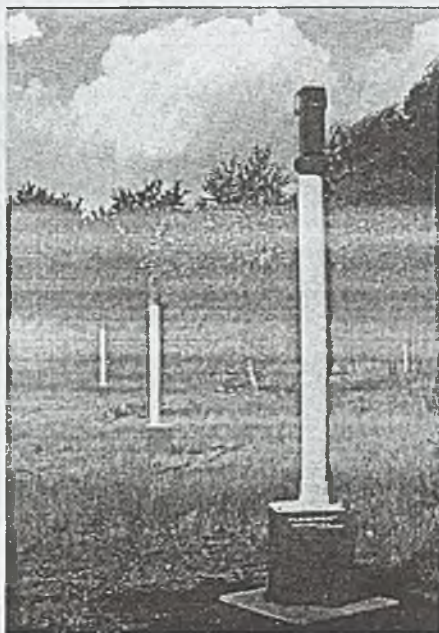
W pracy [16] przedstawiono wykresy pokazujące wpływ dokładności pomiaru transmitancji na zakres pomiarowy transmisometru dla parametru B . Wynika z nich, że rozszerzenie zakresu pomiarowego od góry wiąże się ze zwiększeniem wartości dolnej granicy pomiarowej. Dlatego budowane są zarówno transmisometry z krótką bazą (od 2 m do 20 m) jak i długą (powyżej 25 m). Często stosuje się też konstrukcję mieszaną: w urządzeniu znajdują się dwa odbiorniki, z których jeden umieszczony jest w małej odległości od nadajnika, np. 15 m, a drugi w odległości 75 m, co łącznie umożliwia pomiar widzialności w zakresie od 10 m do 3000 m [27].

Dokładność pomiaru transmisometrów zależy od wielu czynników. Jest istotne by wartość odniesienia F_0 była taka sama przez długi okres czasu, a także by błąd bieżącego pomiaru strumienia F był możliwie mały. W celu zmniejszenia wpływu pierwszego z tych czynników dokonuje się co pewien czas kalibracji przyrządu. Jako przyczyny błędów pomiarów można wskazać [20]: zanieczyszczenie zewnętrznych powierzchni elementów optycznych lub szyb, oddziaływanie warunków zewnętrznych, niestabilność wzajemnego ustawienia układów optycznych nadajnika i odbiornika spowodowana przez ruchy podłoża oraz niestabilność konstrukcji wsporczej nadajnika i odbiornika, zmiany parametrów układów optycznych i elektronicznych.

Aby uniknąć osadzania się pyłów stosuje się specjalne dmuchawy [21] albo też mierzy stopień zanieczyszczenia i stosownie do niego dokonuje się korekty wyniku pomiaru [25]. Mimo to wymagane jest okresowe czyszczenie szyb. Zarówno zanieczyszczenia, jak i nieuniknione zmiany położenia wiązki światła i jej parametrów powodują potrzebę kalibrowania przyrządu. Opracowano też system automatycznej kalibracji transmisometru [20], polegający na wprowadzaniu współczynników korygujących wynik, mierzonych gdy widzialność jest duża ($V_M > 10$ km).

W celu zmniejszenia wpływu innych przyczyn powstawania błędów pomiaru, stosuje się układ automatycznej regulacji strumienia świetlnego wychodzącego z nadajnika, układy zapobiegające wpływowi oświetlenia zewnętrznego, układy zapewniające dużą stabilność wzmocnienia w układzie odbiornika itp.

Osiągnięcia w technice budowy transmisometrów. Do czołowych producentów transmisometrów z długą bazą należą firmy Vaisala (Finlandia) i Jenoptik Impulsphysik (RFN). Transmisometr *Mitras* firmy Vaisala [25] jest przeznaczony do dokonywania pomiarów MOR i RVR na lotniskach kategorii od I do IIIB. W tym ostatnim przypadku stosuje się dwa odbiorniki światła umieszczone w różnej odległości od nadajnika. Odległość maksymalna między nadajnikiem i odbiornikiem wynosi 200 m. Błąd pomiaru transmitancji jest nie większy niż $\pm 1\%$. W urządzeniu zastosowano lampę ksenonową, która emituje światło o długości fali w zakresie 300÷1100 nm (pasmo spektralne odbiornika wynosi 300÷700 nm, maksimum czułości przy 550 nm). Transmisometr *Skopograph II Flamingo* [21] firmy Jenoptik Impulsphysik przeznaczony jest również do instalacji na lotniskach kategorii I-IIIB.



Rys. 2. Transmisometr *Skopograph II Flamingo* firmy Impulsphysik

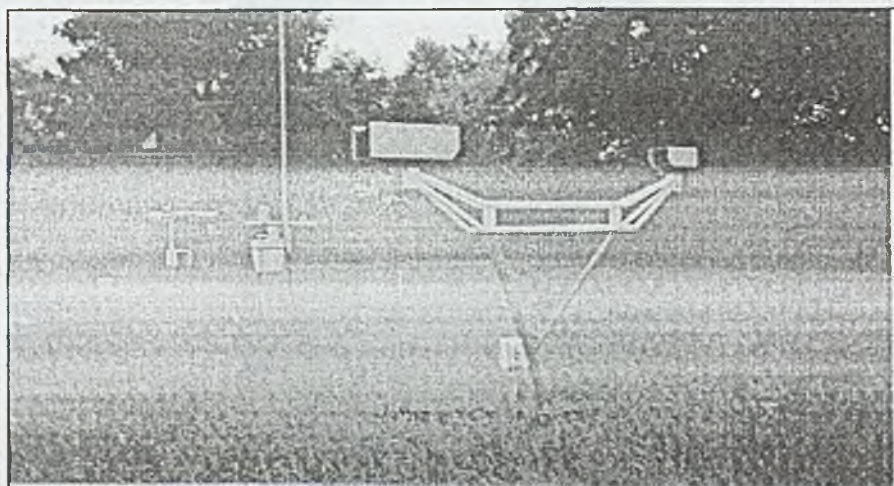
Widok tego urządzenia, złożonego z jednego nadajnika i dwóch odbiorników, umożliwiającego pomiar zasięgu widzialności w zakresie od 15 m do 3000 m przedstawiono na rys. 2.

Uzyskiwane zakresy pomiaru MOR zależnie od odległości bazowej są następujące: 10÷600 m dla $B = 15$ m, 35÷2000 m dla $B = 50$ m, 50÷3000 m dla $B = 75$ m, 10÷3000 m dla bazy 15 m i 75 m (dwa odbiorniki). Jako źródło światła służy lampa ksenonowa pracująca impulsowo z częstotliwością 3 Hz. Odbiornik ma charakterystykę spektralną zbliżoną do charakterystyki czułości oka.

W wytwarzanych obecnie transmisometrach stosowane są zwykle lampy emitujące światło białe. Wyjątkiem jest transmisometr wytwarzany przez firmę Meisei (Japonia), w którym zastosowano diodę elektroluminescencyjną LED wysyłającą promieniowanie podczerwone oraz najnowsze urządzenie firmy GEC-Marconi zawierające również diodę LED, emitującą światło czerwone [24].

W Instytucie Maszyn Matematycznych w Warszawie opracowano transmisometr przeznaczony do oceny zasięgu widzialności horyzontalnej, w którym jako źródło światła zastosowano laser półprzewodnikowy emitujący światło czerwone o długości fali 670 nm [17]. Transmisometr charakteryzuje się małą długością linii bazowej wynoszącą 5 m i zastosowaniem reflektora zawracającego wiązkę światła do odbiornika znajdującego się w jednym zespole konstrukcyjnym z nadajnikiem. Dzięki temu odznacza się małymi, jak na tego rodzaju urządzenia, rozmiarami i masą. Może być instalowany w terenie na odpowiednim fundamencie lub nawet przenośnym podeście.

Na rys. 3 pokazano zdjęcie transmisometru znajdującego się wśród innych przyrządów meteorologicznych na terenie stacji badań i pomiarów Instytutu Meteorologii i Gospodarki Wodnej w Warszawie.



Rys. 3. Transmisometr laserowy TL01 zainstalowany na terenie IMGW

Mała długość linii bazowej umożliwia dokładny pomiar widzialności w warunkach występowania bardzo gęstych mgieł, nawet przy ograniczeniu widzialności do ok. 5 m. Na podstawie pomiarów stabilności pomiaru transmitancji jego górną granicę pomiarową określono na 3000 m, gdy widzialność jest ograniczona z powodu zamgleń. Tak więc uzyskano znacznie szerszy zakres pomiaru widzialności niż było to możliwe w transmisometrach konwencjonalnych z lampą ksenonową przy podobnej długości linii bazowej.

4.2. Skaterometry

Zasada pomiaru. Metoda skaterometryczna polega na pomiarze mocy części promieniowania rozproszonego w powietrzu i wnioskowaniu na tej podstawie, jaka jest wartość współczynnika ekstynkcji ośrodka, a następnie obliczeniu zasięgu widzialności przy założeniu jednorodności ośrodka. Dodatkowym założeniem jest przyjęcie, że udział absorpcji w tłumieniu światła jest pomijalny w stosunku do rozpraszania. Dla takiego przypadku całkowity strumień światła rozproszonego jest równy współczynnikowi ekstynkcji ośrodka dla rozpraszania.

Rozproszenie światła przez cząstkę o pewnej elementarnej objętości można określić za pomocą funkcji rozproszenia kąowego $\beta(\theta, \Phi)$. Jak można wykazać [7], całka tej funkcji dla pełnego kąta bryłowego jest równa

$$\int_{\Omega} \beta(\theta, \Phi) d\omega = \alpha \quad (4.5)$$

przy czym $d\omega$ jest elementarnym kątem bryłowym, a θ, Φ oznaczają współrzędne kątowe.

Ponieważ bezpośredni pomiar całego strumienia rozproszonego nie jest możliwy ze względów technicznych, mierzy się go w obrębie określonego kąta bryłowego. Indykatrysa rozpraszania określona przez funkcję $\beta(\theta, \Phi)$ dla obu rodzajów rozpraszania – molekularnego i aerozolowego wykazuje maksima dla kierunku zgodnego i przeciwnego z kierunkiem rozchodzenia się światła. Zatem istnieją dwa korzystne kierunki pomiaru światła: z tyłu lub z przodu obszaru rozpraszającego. Zależnie od tego przyrządy do pomiaru promieniowania rozproszonego – skaterometry (ang. scatter meters) – dzielą się na mierniki rozpraszania wstecznego (back scatter meters) i rozpraszania do przodu (forward scatter meters).

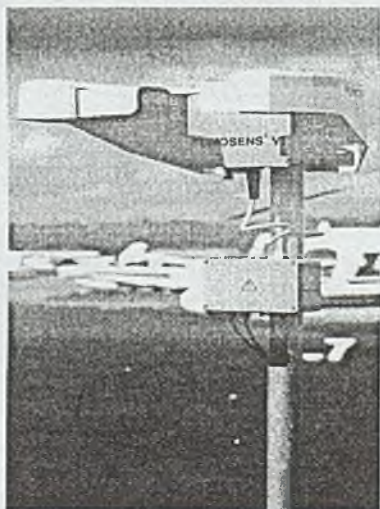
W pierwszym przypadku odbiornik umieszczony jest obok nadajnika i tworzy z nim jeden blok konstrukcyjny, a w drugiej wersji oba zespoły są zamocowane do konstrukcji nośnej oddzielnie. Cenną zaletą skaterometrów jest zwarta budowa, co w połączeniu z łatwością instalacji, niewrażliwością na wstrząsy, a także kilkakrotnie niższą ceną niż transmisometrów powoduje, że ich zastosowanie jest bardzo duże.

Podstawowe cechy metody skaterometrycznej. Główną właściwością metody opartej na pomiarze światła rozproszonego jest możliwość oceny widzialności w bardzo szerokim zakresie pomiarowym – od ok. 10 m do kilkudziesięciu tysięcy metrów. W górnym przedziale zakresu dominującą przyczyną tłumienia światła jest rozpraszanie

Rayleigha, którego korzystną cechą z punktu widzenia pomiarowego jest symetria charakterystyki rozpraszania. Na błąd pomiaru wpływa wtedy głównie wielkość absorpcji molekularnej, z założenia pomijanej przy obliczaniu współczynnika ekstynkcji oraz właściwości aparaturowe – stabilność toru pomiarowego, szumy w układzie elektronicznym, zanieczyszczenia okien.

W dolnym przedziale pomiarowym błąd oceny widzialności jest spowodowany głównie czynnikami środowiskowymi wynikającymi z uzależnienia charakterystyki kątowej rozproszenia od rodzaju mgły lub składu pozostałych aerozoli (pyłów, dymów itd). Indykatrysa rozpraszania aerozolowego jest asymetryczna i stosunek strumienia światła rozproszonego w kierunku propagacji do rozproszonego w kierunku wstecznym zależy od rodzaju i koncentracji cząstek. Tak więc urządzenie zaprogramowane dla określonej charakterystyki rozpraszania ma nieprawidłowe wskazania, jeżeli występują inne przyczyny ograniczenia widzialności. Przykładem może być uzależnienie wyniku pomiaru od rodzaju mgły – radiacyjnej lub adwekcyjnej, które powoduje, że przy pomiarach zasięgów widzialności poniżej 200 m błąd może być bardzo duży [8]. W ostatnich latach podjęto prace nad poprawą dokładności skaterometrów przez jednoczesny pomiar wielu czynników pogodowych (opadów, temperatury itd.) i wnioskowanie przy wykorzystaniu metod eksperckich o rodzaju przyczyny powodującej rozpraszanie światła, a następnie odpowiednie korygowanie wyniku [13]. Stosuje się też zaawansowane metody pomiaru sygnału optycznego [9].

Osiągnięcia w technice budowy skaterometrów. Wśród wytwarzanych seryjnie skaterometrów można wyróżnić grupę skaterometrów o dużych zakresach pomiarowych, mierzących rozproszenie do przodu i przeznaczonych dla potrzeb komunikacji lotniczej oraz drugą grupę przyrządów do zastosowań w komunikacji drogowej i morskiej, w których wykorzystuje się obie metody pomiaru rozpraszania, o małych zakresach pomiaru widzialności i mniejszej dokładności pomiaru.



Rys. 4. Skaterometr *FUMOSENS VI* firmy Impulsphysik

Jako przykład realizacji skaterometru należącego do pierwszej grupy można wskazać przyrząd *FUMOSENS VI* firmy Jenoptik Impulsphysik [22], który przedstawiono na rys. 4.

Urządzenie składa się z dwu nadajników i dwu odbiorników światła, pracujących na przemian, przy czym jednocześnie mierzone jest światło przechodzące i światło rozproszone pod kątem 35° . Dzięki temu można uwzględnić wpływ na wynik pomiaru zanieczyszczenia okien i zmian wzmocnienia sygnału. Zakres pomiaru widzialności wynosi od 20 m do 20 km. Jako źródło światła zastosowano diodę emitującą światło czerwone. Odbiornikami są fotodiody zaopatrzone w filtr wąskopasmowy. Jako główne zastosowanie wskazuje się pomiar widzialności na drogach, lotniskach i stacjach meteorologicznych.

5. Zastosowanie metod analizy obrazu do oceny zasięgu widzialności

Omówione poprzednio metody pośredniego obliczania zasięgu widzialności, aczkolwiek umożliwiają stosunkowo dokładną ocenę przezroczystości powietrza w badanym miejscu, nie zapewniają uzyskania wystarczającej informacji o widzialności obiektów rzeczywistych. Dlatego w dalszym ciągu trwają prace nad urządzeniami działającymi na zasadzie fotometrycznego pomiaru kontrastu obserwowanych obiektów i tła. Rozwój techniki, a w szczególności metod pozyskiwania i analizy obrazu, stwarza szansę automatycznego i dokładnego pomiaru tego parametru. Przyrządy, które powstały dla potrzeb monitorowania widzialności w komunikacji lotniczej można podzielić na dwie grupy [14]:

- automatyczne analizatory widzialności światła na pasie startowym (wideometry liniowe),
- automatyczne analizatory widzialności czarnych obiektów w obszarze lotniska (wideometry przestrzenne).

Do pierwszej grupy zalicza się przyrząd opisany w pracy [5], który jest przeznaczony do pomiaru parametru RVR. Jego sensorem jest specjalna kamera, umieszczona na wieży w pozycji odpowiadającym pozycji pilota samolotu zbliżającego się do lądowania, a obiektem mierzonym rząd lamp rozmieszczonych równoległe do pasa. Kamera pokazuje obraz kolejno włączanej grupy trzech lamp o znanej odległości od kamery. Odległość najbardziej oddalonego źródła światła, które obsługujący urządzenie jeszcze może zauważyć, jest uznawana za wielkość RVR. Urządzenie zostało też przystosowane do automatycznego pomiaru RVR. W tym celu zastosowano pojedynczy fotodetektor i mierzono strumień tylko jednej z kolejno włączanych lamp. Czulość detektora była zmieniana w zależności od luminancji tła w sposób odpowiadający zmianie czulości oka. W celu eliminacji sygnałów zakłócających pochodzących od oświetlenia tła, szumu i innych sygnałów, strumień światła jest modulowany. Układ optyczny detektora ma regulowaną ogniskową, w celu dostosowania go do zmieniającej się odległości źródła światła od kamery. Jak wykazały badania uzyskano dużą zgodność mierzonych wyników z rezultatami obserwacji wizualnej. Twórcy przyrządu oceniają, że można za jego pomocą mierzyć RVR w zakresie 30-1200 m i MOR od 1200 m do 10 km.

Jeśli chodzi o wideometry przestrzenne, to tego rodzaju system opisano w pracy [3]. Jego głównymi składnikami są kamera umieszczona na stole obrotowym oraz analizator obrazu. Kamera służy do dostarczania obrazów otoczenia na tle horyzontu dla wybranych kątów azymutu. Obrazy te są rejestrowane i analizowane w celu określenia zasięgu widzialności. Przedmiotem analizy jest obecność na obrazie określonych stałych obiektów, charakteryzujących się możliwie dużym pochłanianiem światła, których odległość od kamery jest znana. Oprócz pomiaru luminancji tych obiektów mierzona jest również średnia jasność nieba przy horyzoncie. W rezultacie odpowiednich obliczeń uzyskuje się wartość pozornego kontrastu obiektów i określa zasięg widzialności zgodnie z definicją podaną we wstępie do artykułu. Przy użyciu złożonych algorytmów pozwala to z kolei na obliczenie uśrednionego zasięgu widzialności w sektorze i ogólnego zasięgu widzialności. Do pomiarów luminancji w warunkach nocnych można zastosować układ stałych źródeł światła o znanym natężeniu.

6. Zakończenie

Mimo dużego postępu, jaki nastąpił w ostatnich latach w dziedzinie techniki pomiarów zasięgu widzialności, trudno uznać istniejący stan za zadowalający. Będące w użyciu systemy polegające na pomiarze transmitancji ośrodka lub rozpraszania światła określają widzialność obiektów o specyficznych cechach (ciał doskonale czarnych) znajdujących się w ośrodku jednorodnym, przy założeniu pewnego standardowego progu czułości kontrastowej. Systemy oparte na pomiarze kontrastu obiektów, mimo że mierzą zasięg widzialności obiektów rzeczywistych nie znalazły jeszcze szerszego zastosowania, głównie z powodu trudności w zapewnieniu dokładnego wyniku pomiaru. Wydaje się, że istotne zwiększenie wiarygodności i dokładności pomiaru widzialności można by uzyskać przez budowę systemu multisensorowego, zawierającego różnego rodzaju czujniki fotoniczne. Spośród nich sensor działający na zasadzie analizy obrazu obserwowanych obiektów mógłby np. dostarczać podstawowej informacji odnoszącej się do zasięgu widzialności tych obiektów, a pozostałe wprowadzałyby poprawki korygujące wynikające z pomiarów przezroczystości powietrza, oświetlenia ogólnego, czynników meteorologicznych itd. Budowa takiego systemu wiązałaby się z koniecznością opracowania specjalnego oprogramowania uwzględniającego powiązania między mierzonymi parametrami.

Literatura

- [1] Blackwell H. R., *Contrast Thresholds of the Human Eye*, J. Opt. Soc. Am., Vol. 36, No. 6, 1946, 624-643.
- [2] Boreux J.-J., Duckstein L., *A fuzzy approach to the definition of standardized visibility in fog*, Appl. Mathematics a. Computation, Vol. 61, No. 2-3, 1994, 287-299.
- [3] Brown H. A., Burnham D. C., *Evaluation of Prevailing Visibility Sensor Based on a Scanning Solid State Camera*, Eight Symposium on Meteorological Observations a. Instrumentation, Amer. Met. Soc. Anaheim, 1993, 80-85.
- [4] Dickson D. R., Hales J. V., *Computation of Visual Range in Fog and Low Clouds*, Journal of Applied Meteorology, Vol. 2, April 1963, 281-285.

- [5] Etienne M., De Swert C., *Automatic Videometer for the Runway Visual Range Measurement*, Instr. and Observ. Met. Rep. No. 15, Geneva, 1984, 151-155.
- [6] Gavrilov V. A., *Vidimost' v atmosfere*, Gidrometeorologicheskoe Izdatel'stvo, Leningrad, 1966.
- [7] Grigges D. J. et al., *The First WMO Intercomparison of Visibility Measurements*, WMO Report No. 41, 1989.
- [8] Hahn W. et al., *Visibility Measurements of Fog on Highways Using a Lidar Method in a Car*, Laser in der Umweltmeßtechnik, Vorträge des 11. Internationalen Kongresses Laser 93, Springer-Verlag, 1993, 173-176.
- [9] Holejko K., Nowak R., *Fog detection and visibility measurements using forward scatter radiation*, Opto-Electr. Rev., Vol. 5, No. 2, 1997, 107-110.
- [10] Koschmieder H., *Theorie der horizontalen Sichtweite*, Beitr. Phys. Atm., XII, 1925, 33-55.
- [11] Kruse P. W., McGlauchin L. D., McQuistan R. B., *Elements of infrared technology: generation, transmission and detection*, J. Wiley and Sons, Inc. New York, 1962.
- [12] Middleton W. E. K., *Vision Through the Atmosphere*, University of Toronto Press, Toronto, 1952.
- [13] Noormohammadian M., Oppel U.G., Starkov A.V., *Expertensysteme zur Datenanalyse technischer und ökologischer Multisensorsysteme*, Laser in der Umweltmeßtechnik, Vorträge des 11. Internationalen Kongresses Laser 93, Springer-Verlag, 1993, 190-197.
- [14] Rózdżyński K., *Miernictwo meteorologiczne*, Wyd. IMiGW, Warszawa, 1996.
- [15] Synak R., *Ocena widzialności w powietrzu zamglonym metodą pomiaru transmisji światła monochromatycznego*, Techniki Komputerowe, IMM, Nr 1, 1994, 5-28.
- [16] Synak R., *Laser Transmissometer*, Laser Technol. IV, Proc. SPIE, Vol. 2202, 1995, 384-388.
- [17] Synak R. et al., *Transmissometr laserowy TL 01 - budowa i wyniki badań*, Techniki Komputerowe, IMM, Nr 1, 1996, 49-64.
- [18] Zuyev V. E., *Laser-Light Transmission Through the Atmosphere*, Laser Monitoring of the Atmosphere, Hinkley E.D. - ed., Springer-Verlag New York Inc., 1976, 3.1-3.69.
- [19] ICAO, *Meteorological Service for International Air Navigation, Annex 3 to the Conv. on Int. Civil Av.*, July 1995.
- [20] IMPULSPHYSIK, *Autocalibration System for International Airports*, JENOPTIK IMPULSPHYSIK GmbH, Schenefeld, 1996.
- [21] IMPULSPHYSIK, *Get your CAT Flamingo Skopograph II*, JENOPTIK IMPULSPHYSIK GmbH, Schenefeld, 1997.
- [22] IMPULSPHYSIK, *The Twins. FUMOSENS VI, STILBUS II, Forward Scatter Instruments*, JENOPTIK IMPULSPHYSIK GmbH, Schenefeld, 1997.

-
- [23] IMPULSPHYSIK, FUMOSENS E+D85, JENOPTIK IMPULSPHYSIK GmbH, Schenefeld, 1998.
 - [24] Marconi, Airport Visibility Monitoring, GEC-Marconi Radar&Defence Systems, Leicester, 1998.
 - [25] VAISALA, MITRAS Transmissometer - Reliable and Accurate, VAISALA Oy, Helsinki, <http://www.vaisala.com/WWW/webcom.NSF>, 1999.
 - [26] VAISALA, PWD11 Present Weather Detector, VAISALA Oy, Helsinki, <http://www.vaisala.com/WWW/webcom.NSF>, 1999.
 - [27] VAISALA, FD12P Weather Sensor - Combined Visibility and Present Weather Sensor, VAISALA Oy, Helsinki, <http://www.vaisala.com/WWW/webcom.NSF>, 1999.

DARIUSZ CABAN
INSTYTUT INFORMATYKI TEORETYCZNEJ
I STOSOWANEJ PAN GLIWICE

8-bitowe mikrosterowniki z pamięcią FLASH programowalne przez synchroniczny interfejs szeregowy SPI

8-bit microcontrollers with FLASH memory programmable through SPI synchronous serial interface

Streszczenie

W artykule przedstawiono mikrosterowniki z pamięcią programu typu FLASH, programowalne również w sposób szeregowy za pośrednictwem interfejsu SPI. Opcja programowania szeregowego umożliwia zmianę zawartości pamięci wewnętrznej bez potrzeby usuwania mikrosterownika z systemu docelowego. Omówiono sposób programowania tych mikrosterowników. Zaprezentowano także sposób programowej emulacji interfejsu SPI przy wykorzystaniu portu drukarki mikrokomputera IBM PC.

Abstract

This article presents microcontrollers with FLASH program memory, programmable also serially through SPI interface. Serial programming option enables program downloading without need to remove microcontroller from a target system. Methods of programming of these microcontrollers are described. Also, method of software emulation of SPI interface using printer port of IBM PC microcomputer is presented.

1. Wprowadzenie

Do realizacji wewnętrznej pamięci programu w mikrosterownikach zaczęto w ostatnich latach stosować także pamięć FLASH. Zarówno zapis jak i kasowanie takiej pamięci odbywa się za pomocą impulsów elektrycznych. Dzięki temu mikrosterownik z pamięcią FLASH może być umieszczony w obudowie plastikowej, przez co jest on tańszy od swojego odpowiednika zawierającego pamięć EPROM w wersji wielokrotnie programowalnej, który musi posiadać obudowę ceramiczną z okienkiem.

Ze względu na sposób programowania mikrosterowniki z pamięcią FLASH można podzielić na dwie grupy:

- programowalne tylko równolegle,
- programowalne równolegle oraz opcjonalnie szeregowo.

Opcja programowania szeregowego jest szczególnie użyteczna dla konstruktorów – zmiana zawartości wewnętrznej pamięci programu może zostać przeprowadzona również po zainstalowaniu mikrosterownika w systemie docelowym (jest to tzw. programowanie w systemie). Większość takich mikrosterowników jest programowana poprzez wbudowany w nie interfejs transmisji asynchronicznej, bezpośrednio z mikrokomputera np. IBM PC. Rzadziej wykorzystuje się interfejsy synchroniczne. W tej grupie ciekawymi rozwiązaniami są mikrosterowniki z interfejsem SPI (*Serial Peripheral Interface*), które mogą być programowane również bezpośrednio z mikrokomputera IBM PC; zostały one przedstawione w niniejszej publikacji.

2. Interfejs SPI

Synchroniczny interfejs szeregowy SPI został opracowany przez firmę Motorola dla realizacji połączeń pomiędzy jednostką centralną systemu mikrokomputerowego a wieloma układami zewnętrznymi. Jednostka centralna jest w interfejsie układem nadrzędnym (*master*) w stosunku do pozostałych, nazywanych układami podrzędnymi (*slaves*). Informacja jest przesyłana za pośrednictwem trzech linii (w nawiasach podano stosowane ich oznaczenia):

- danych wejściowych dla układów podrzędnych (MOSI – *Master Out Slave In*, DI - *Data In*, SI - *Slave In*),
- danych wyjściowych z układów podrzędnych (MISO – *Master In Slave Out*, DO - *Data Out*, SO - *Slave Out*),
- impulsów zegarowych (SCK, SCLK – *Serial Clock*).

Ponadto do każdego układu podrzędnego doprowadzana jest osobna linia wyboru (CS – *Chip Select*, SS - *Slave Select*), w większości przypadków stanem aktywnym tej linii jest stan niski. Do zadań układu nadrzędnego należy inicjacja i sterowanie przebiegiem komunikacji z wybranym układem podrzędnym oraz wytwarzanie impulsów zegarowych, niezależnie od tego, czy w danej chwili jest nadawcą czy odbiorcą informacji. Bity informacyjne są w interfejsie SPI synchronizowane zboczami impulsów zegarowych ([2]).

Sprzętowy interfejs SPI jest elementem struktury wewnętrznej niektórych mikrosterowników, natomiast nie są produkowane takie interfejsy w postaci osobnych układów scalonych. Realizację sprzętową interfejsu można też zastąpić realizacją programową. Jest to możliwe z uwagi na to, że wytwarzanym wyłącznie przez układ nadrzędny impulsom zegarowym nie stawia się rygorystycznych wymagań czasowych – w trakcie transmisji może się zmieniać ich częstotliwość oraz wypełnienie. Dzięki temu układy zewnętrzne wyposażone w interfejs SPI mogą współpracować z mikrosterownikiem dowolnego typu.

3. Mikrosterowniki programowalne przez interfejs SPI

3.1. Mikrosterownik AT89S8252

Mikrosterownik AT89S8252 jest produkowany przez firmę Atmel i stanowi rozbudowaną wersję układu 80C52, należącego do rodziny MCS51 ([3],[6]). Mikrosterowniki tej rodziny są bardzo popularne i istnieje na ich temat obszerna literatura, stąd ograniczymy się tu do występujących w AT89S8252 rozszerzeń w stosunku do pierwowzoru. Oprócz interfejsu SPI mikrosterownik AT89S8252 zawiera następujące rozszerzenia:

- 8 kB pamięci programu FLASH,
- 2 kB pamięci danych EEPROM,
- dodatkowy rejestr wskaźnikowy DPTR,
- dodatkowe funkcje układu zegara/licznika T2,
- układ nadzorczy (*watchdog*),
- możliwość blokowania generacji sygnału ALE,
- częstotliwość zegarowa w zakresie 0÷24 MHz,
- napięcie zasilania 4÷6 V.

Gwarantowana przez producenta trwałość pamięci FLASH wynosi 1000 przeprogramowań, natomiast pamięci EEPROM – 100000. Pamięć EEPROM współdzieli obszar o adresach 0÷7FF z zewnętrzną pamięcią danych. Dostęp do obu rodzajów pamięci jest realizowany za pomocą rozkazu MOVX. Rozróżnienie pamięci przy odwołaniach następuje poprzez odpowiednie ustawienie jednego z bitów sterujących w obszarze rejestrów specjalnego przeznaczenia.

Jeżeli układ zegara/licznika T2 pracuje jako generator impulsów taktujących dla standardowego układu transmisji szeregowej, wówczas istnieje możliwość wyprowadzenia na wyjście P1.0 fali prostokątnej o wypełnieniu 1/2 i częstotliwości zależnej od częstotliwości zegarowej mikrosterownika i ustawionej prędkości transmisji.

Układ nadzorczy mikrosterownika AT89S8252 został zrealizowany jako układ zliczający impulsy wytwarzane przez niezależny wewnętrzny generator współpracujący z programowalnym podzielnikiem częstotliwości. Nominalna częstotliwość generatora wynosi 62.5 Hz, a współczynnik podziału ma postać 2^n , gdzie $n = 0, 1, \dots, 7$, co daje zakres maksymalnych czasów odświeżania układu nadzorczego od 16 do 2048 ms. W rzeczywistości występują odchylenia wartości maksymalnego czasu odświeżania od wartości zaprogramowanej dochodzące do $\pm 30\%$.

Wyprowadzeniami interfejsu SPI są te linie portu P1, którym w układzie 80C52 nie przydzielono funkcji alternatywnych. Interfejs ten można wykorzystywać również w trybie normalnej pracy mikrosterownika do komunikacji z otoczeniem; mikrosterownik może być w tym trybie układem nadrzędnym lub podrzędnym. W trybie programowania AT89S8252 jest traktowany jako układ podrzędny. Dla każdego z wymienionych przypadków inne są funkcje wyprowadzonych na zewnątrz mikrosterownika linii interfejsu SPI, co pokazano w tabeli 1.

Tabela 1. Funkcje linii interfejsu SPI mikrosterownika AT89S8252 w różnych trybach pracy

linia interfejsu SPI mikrosterownika	tryb programowania mikrosterownika	tryb normalnej pracy mikrosterownika	
		jako układ nadrzędny	jako układ podrzędny
\overline{SS} (P1.4)	nieużywana	nieużywana	wybór układu
MOSI (P1.5)	wejście	wyjście	wejście
MISO (P1.6)	wyjście	wejście	wyjście
SCK (P1.7)	wejście	wyjście	wejście

Częstotliwość zegarowa programowanego szeregowo mikrosterownika AT89S8252 powinna mieścić się w zakresie 4÷24 MHz, a częstotliwość impulsów zegarowych przesyłanych linią SCK nie może przekroczyć 1/40 częstotliwości zegarowej. Do programowania służy siedem 3-bajtowych rozkazów, których format przedstawiono w tabeli 2. Bajt pierwszy i drugi są przesyłane linią MOSI i reprezentują kod rozkazu, natomiast trzeci bajt zawiera daną, przy operacji odczytu przesyłaną linią MISO. Bity danych wejściowych są wpisywane do mikrosterownika w chwilach wyznaczonych przez narastające zbocza impulsów zegarowych, natomiast kolejne bity odczytywanej informacji są przez niego wystawiane na linię danych wyjściowych przy zboczach opadających.

Tabela 2. Lista rozkazów programowania szeregowego mikrosterownika AT89S8252

Rozkaz	Format rozkazu		
	bajt 1	bajt 2	bajt 3
rozpoczęcie programowania	10101100	01010011	xxxxxxxx
kasowanie całej zawartości FLASH oraz EEPROM	10101100	xxxxx100	xxxxxxxx
odczyt bajtu pamięci FLASH	$a_1a_2a_{11}a_{10}a_9a_8001$	$a_7a_6a_5a_4a_3a_2a_1a_0$	$d_7d_6d_5d_4d_3d_2d_1d_0$
zapis bajtu pamięci FLASH	$a_1a_2a_{11}a_{10}a_9a_8010$	$a_7a_6a_5a_4a_3a_2a_1a_0$	$d_7d_6d_5d_4d_3d_2d_1d_0$
odczyt bajtu pamięci EEPROM	$00a_{10}a_9a_8101$	$a_7a_6a_5a_4a_3a_2a_1a_0$	$d_7d_6d_5d_4d_3d_2d_1d_0$
zapis bajtu pamięci EEPROM	$00a_{10}a_9a_8110$	$a_7a_6a_5a_4a_3a_2a_1a_0$	$d_7d_6d_5d_4d_3d_2d_1d_0$
programowanie bitów zabezpieczających	10101100	$lb_1lb_2lb_3xx111$	xxxxxxxx
a_i – bity adresowe d_i – bity danych	lb_i – bity zabezpieczające (programowane, jeśli $lb_i = 0$) x – wartość bitu nieistotna		

Przejdzie w tryb programowania jest wymuszany poprzez ustawienie na wejściu zerowania RST mikrosterownika logicznego stanu wysokiego i wysłanie rozkazu rozpoczęcia programowania. Zapisywany bajt, zarówno pamięci FLASH jak i EEPROM, jest najpierw automatycznie kasowany. Dzięki temu możliwa jest również zmiana zawartości tylko wybranego fragmentu pamięci. Maksymalny czas zapisu bajtu wynosi 2.5 ms przy napięciu zasilania mikrosterownika równym 5V. Można uzyskać informację o stanie zaawansowania tej operacji – w tym celu należy dokonać odczytu bajtu i porównać odczytywaną wartość z zapisywaną. Pozytywny wynik porównania wskazuje na to, że zapis bajtu został zakończony. Chociaż nie jest konieczne kasowanie całej pamięci przed rozpoczęciem zapisu, konstruktorzy mikrosterownika AT89S8252 przewidzieli i taki rozkaz. Kasowanie zajmuje ok. 16 ms, próba odczytu w tym czasie dowolnego bajtu da w wyniku wartość 00H. Zakończenie programowania mikroste-

rownika następuje poprzez ponowne ustawienie na jego linii zerowania logicznego stanu niskiego, po czym mikrosterownik rozpocznie wykonywanie programu począwszy od adresu 0000H.

Do transmisji rozkazów podczas programowania mikrosterownika AT89S8252 można w prosty sposób wykorzystać standardowy port drukarki mikrokomputera IBM PC ([1]). Port drukarki jest obsługiwany za pośrednictwem bloku trzech rejestrów umieszczonych w przestrzeni adresowej układów wejścia-wyjścia mikrokomputera. Są to rejestry: danych, statusu oraz sterujący. Połączenie mikrokomputera z programowanym mikrosterownikiem wymaga trzech linii wyjściowych i jednej linii wejściowej portu. Przyjmując, że sygnał PROG zmiany trybu pracy mikrosterownika będzie generowany na linii DATA 4 oraz takie przyporządkowanie linii portu liniom interfejsu SPI mikrosterownika:

```
DATA 5   →  MOSI
DATA 6   →  SCK
ACK/     ←  MISO
```

operacje nadawania i odbioru bajtu zgodnie z protokołem interfejsu SPI można zakodować w sposób następujący np. przy zastosowaniu języka Turbo C:

```
#include <dos.h>

#define uchar  unsigned char
#define DATA  0x0378      /* adres bazowy rejestrów portu */
#define STATUS DATA+1

uchar output;             /* zawartość rejestru danych portu */

void send(uchar byte)
{
    uchar counter = 8;

    while(counter--)
    {
        if (byte & 0x80)
            output |= 0x20;          /* MOSI ←- 1 */
        else
            output &= 0xDF;         /* MOSI ←- 0 */
        outportb(DATA, output);
        wait();
        outportb(DATA, output | 0x40); /* SCK ←- 1 */

        wait();
        outportb(DATA, output);      /* SCK ←- 0 */
        byte <<= 1;
    }
}

uchar receive(void)
```

```

{
uchar counter = 8,
    byte = 0,

    while(counter--)
    {
byte <<= 1;
    if (inportb(STATUS) & 0x40)
        byte |= 1;
    outportb(DATA, output | 0x40);    /* SCK <- 1 */
    wait();
    outportb(DATA, output);          /* SCK <- 0 */
    wait();
    }
return(byte);
}

```

Przed rozpoczęciem programowania mikrosterownika trzeba wykonać instrukcję:

```
outportb(DATA, output = 0x8F);    /* SCK = MOSI = PROG = 0 */
```

Należy zagwarantować odpowiednie parametry czasowe impulsów SCK – ich częstotliwość nie powinna przekraczać 1/40 częstotliwości zegarowej programowanego mikrosterownika, która z kolei musi mieścić się w zakresie 4÷24 MHz. W podanym przykładzie jest to realizowane za pomocą funkcji `wait()`, której zadaniem jest odmierzenie czasów trwania, o wartościach rzędu mikrosekund, poszczególnych stanów logicznych pojedynczego impulsu¹. Odmierzanie tych czasów w sposób niezależny od szybkości procesora mikrokomputera IBM PC umożliwi wbudowany układ czasowy 8254 [0]. Układ ten zawiera trzy niezależnie programowane liczniki, oznaczone cyframi 0-2, taktowane wspólnym sygnałem zegarowym o częstotliwości 1.19318 MHz. Licznik 0 jest wykorzystywany przez system operacyjny mikrokomputera, licznik 1 służy do odświeżania pamięci dynamicznej, a licznik 2 jest używany do generowania sygnału dźwiękowego. Generację sygnału dźwiękowego można jednak wyłączyć, co pozwala na zastosowanie licznika 2 także do innych celów. Dla realizacji odmierzenia czasu należy go zaprogramować w sposób następujący:

```
outportb(0x43, 0x90);    /* tryb 0, zapisywany tylko LSB */
oraz ustawić sygnał bramkujący GATE2 licznika za pomocą instrukcji:
outportb(0x61, inportb(0x61) | 0x01);
```

Postać funkcji `wait()` jest następująca:

```

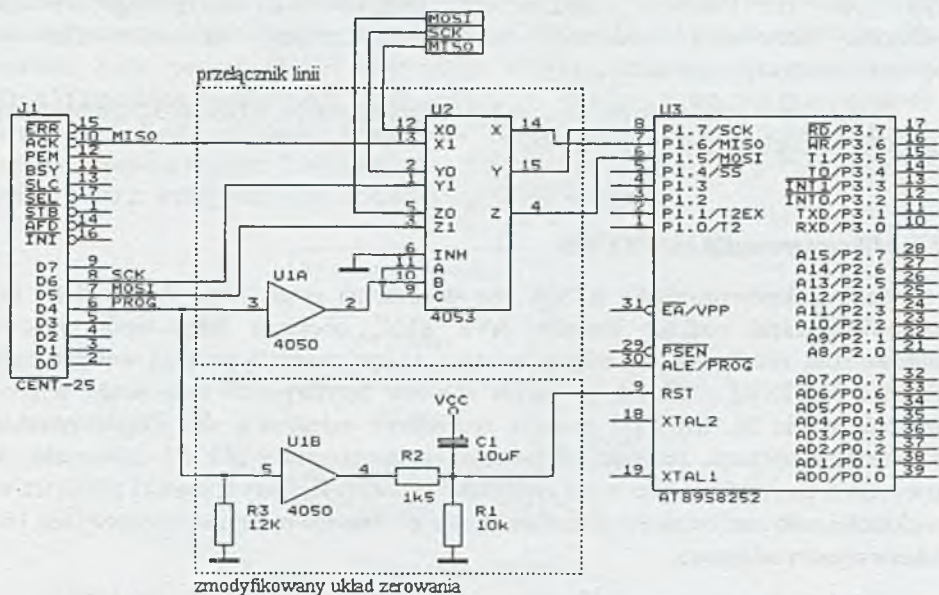
void wait(void)
{
    outportb(0x42, 0x05);    /* zainicjowanie odmierzenia */
    while(!(inportb(0x62) & 0x01));    /* dopóki OUT2 == 0 */
}

```

¹ zastosowanie do tego celu bibliotecznej funkcji `delay()` byłoby rozwiązaniem wysoce nieefektywnym – funkcja ta odmierza czas z rozdzielczością 1 ms.

Zapisanie wartości 0x05 do rejestru licznika 2 gwarantuje, że wartość odmierzonego czasu wyniesie minimum 5.03 μ s, zatem częstotliwość impulsów SCK nie przekroczy 100 kHz. Umożliwi to zaprogramowanie mikrosterownika AT89S8252 taktowanego sygnałem zegarowym o dowolnej częstotliwości z dopuszczalnego zakresu.

Zaprojektowanie i uruchomienie kompletnego programu ładującego (*loader*), na podstawie informacji zawartych w ([1], [3]), nie jest dla użytkownika zadaniem trudnym, ale czasochłonnym. Alternatywą jest użycie jednego z gotowych takich programów, udostępnianych bezpłatnie w sieci Internet (np. *AT89S8252 In System Programmer V1.00*, opracowany przez firmę AEC Electronics, <http://www.aec-electronics.co.nz/>).



Rys. 1. Przełącznik linii interfejsu SPI mikrosterownika pomiędzy układami zewnętrznymi a portem drukarki mikrokomputera IBM PC oraz zmodyfikowany układ zerowania mikrosterownika

Jeżeli planowane jest wykorzystanie linii interfejsu SPI mikrosterownika w systemie docelowym, wówczas konieczne staje się ich przełączanie między mikrokomputerem programującym a układami zewnętrznymi. Funkcje tych linii zmieniają się w zależności od aktualnego trybu pracy mikrosterownika (tabela 1), stąd przełącznik powinien umożliwiać dwukierunkowy przepływ sygnałów. Przykładową realizację takiego przełącznika przedstawiono na rysunku 1. Wykorzystano w niej niedrogi i łatwo dostępny układ CMOS typu 4053, zawierający trzy niezależnie adresowane 1-bitowe multipleksery/demultipleksery analogowe ([5]). Na rysunku przedstawiono również zmodyfikowany układ zerowania mikrosterownika. Twórcy wymienionego wyżej programu ładującego zalecają, aby sygnał PROG zmiany trybu

pracy mikrosterownika był podawany z portu drukarki poprzez opornik 1.5kΩ. Przeprowadzone próby wykazały jednak, że taka modyfikacja jest niewystarczająca – przy PROG = 1 poziom napięcia na wejściu RST mikrosterownika był niższy od minimalnej wartości potrzebnej do zmiany trybu. Poprawę uzyskano po zastosowaniu dodatkowej bramki buforowej 4050 ([5]). Wejście tej bramki przyłączono poprzez opornik do masy, przez co po zakończeniu programowania stan niski na wejściu zerowania mikrosterownika zostanie utrzymany również po odłączeniu mikrokomputera programującego od systemu docelowego. Projektant oprogramowania użytkowego musi też w tym przypadku zadbać o to, aby mikrosterownik nie realizował operacji na liniach interfejsu SPI przez czas co najmniej 500ms po wyzerowaniu. Naruszenie tej zasady spowoduje, że mikrosterownik będzie mógł być zaprogramowany w sposób szeregowy tylko jeden raz. Ponowne włączenie opcji programowania szeregowego wymaga wówczas skasowania zawartości wewnętrznej pamięci mikrosterownika w specjalizowanym programatorze ([3]).

Firma Atmel produkuje również wersję mikrosterownika AT89S8252 dostosowaną do zasilania napięciem z zakresu 2.7÷6 V.

3.2. Mikrosterowniki serii AT90S

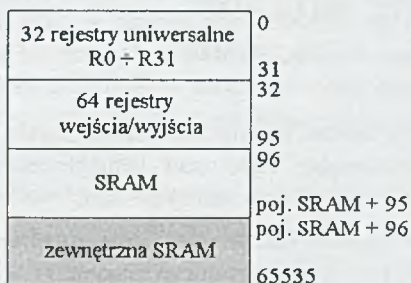
Serię mikrosterowników AT90S wprowadziła na rynek firma Atmel. Seria ta, stanowiąca część rodziny układów AVR RISC, obejmuje kilka typów mikrosterowników, różniących się między sobą m. in. pojemnością pamięci wewnętrznej: FLASH, EEPROM i SRAM, zestawem układów peryferyjnych oraz liczbą wyprzedań (tabela 3). Ich CPU posiada architekturę rejestrową, do przechowywania argumentów operacji, zarówno adresów jak i danych, służy plik 32 uniwersalnych rejestrów 8-bitowych, oznaczonych symbolami R0÷R31. Rejestry R26÷R31 pełnią też w większości mikrosterowników dodatkowe funkcje – łączone parami są używane jako 16-bitowe rejestry adresowe.

Tabela 3. Zestawienie zasobów wewnętrznych mikrosterowników serii AT90S

Typ	L. w. ¹	FLASH ²	EEPROM ³	SRAM ³	Układy wejścia/wyjścia
AT90S2323	8	1024	128	128	licznik 8-bitowy, układ nadzorczy
AT90S2343	8	1024	128	128	
AT90S1200	20	512	64	brak	licznik 8-bitowy, komparator analogowy, układ nadzorczy
AT90S2313	20	1024	128	128	liczniki: 8- i 16-bitowy, komparator analogowy, układ nadzorczy, UART
AT90S2333	28	1024	128	128	liczniki: 8- i 16-bitowy, komparator analogowy,
AT90S4433	28	2048	256	128	układ nadzorczy, UART, SPI ⁴ , 6-kanalowy przetwornik A/C
AT90S4414	40	2048	256	256	liczniki: 8- i 16-bitowy, komparator analogowy,
AT90S8515	40	4096	512	512	układ nadzorczy, UART, SPI

AT90S4434	40	2048	256	256	liczniki: dwa 8- i 16-bitowy, komparator analogowy, układ nadzorczy, UART, SPI, 8- kanałowy przetwornik A/C
AT90S8535	40	4096	512	512	
1) liczba wyprowadzeń 2) pojemność pamięci podana w słowach 16-bitowych 3) pojemność pamięci podana w bajtach 4) oznacza to, że interfejs SPI można wykorzystywać także w trybie normalnej pracy mikrosterownika do realizacji komunikacji z otoczeniem					

Przestrzenie adresowe pamięci FLASH, EEPROM i SRAM są rozdzielone. Dołączenie zewnętrznej pamięci programu jest niemożliwe². W przestrzeni adresowej zajmowanej przez pamięć SRAM umieszczono również plik rejestrów uniwersalnych R0÷R31 oraz blok rejestrów wejścia/wyjścia; organizację tej przestrzeni przedstawiono na rysunku 2. Dostęp do pamięci EEPROM jest realizowany za pośrednictwem trzech rejestrów wejścia/wyjścia. Gwarantowana przez producenta trwałość pamięci FLASH wynosi 1000 przeprogramowań, a pamięci EEPROM 100000.



□ - tylko w AT90S4414 i AT90S8515

Rys. 2. Organizacja przestrzeni adresowej pamięci SRAM w mikrosterownikach serii AT90S

Lista rozkazów liczy 118 pozycji, przy czym w niektórych mikrosterownikach jest ona ograniczona do 89. Większość rozkazów jest wykonywanych w jednym cyklu zegara; czas trwania cyklu jest równy odwrotności częstotliwości zegarowej. W zależności od typu mikrosterownika dostępnych jest 5÷11 trybów adresowania.

Mikrosterowniki serii AT90S są wyposażone w 1÷3 liczniki:

- 8-bitowy T/C0,
- 16-bitowy T/C1,
- 8-bitowy TC/2.

² dotyczy to wszystkich typów mikrosterowników serii AT90S – w każdym z nich długość licznika rozkazów jest równa $\log_2(\text{pojemność pamięci FLASH})$

Liczniki T/C0 i TC/1 mogą zostać zaprogramowane do pracy w trybie odmierzenia czasu lub zliczania impulsów zewnętrznych, podawanych na wejścia licznikowe T0 i T1. Przy pracy w trybie odmierzenia czasu licznik jest taktowany impulsami o częstotliwości f_{CK}/n_1 , gdzie f_{CK} jest częstotliwością zegarową mikrosterownika, a n_1 dobiera się ze zbioru wartości: 1, 8, 64, 256, 1024. Przy zliczaniu impulsów zewnętrznych zmiana zawartości licznika może następować przy narastającym lub opadającym zboczu. Licznik T/C1 dostosowany jest jeszcze do pracy w następujących trybach:

- porównywania,
- przechwytywania,
- modulacji szerokości impulsów.

Licznik T/C2 występuje tylko w mikrosterownikach AT90S4434 i AT90S8535. Licznik ten jest taktowany impulsami o częstotliwości f_{PCK2}/n_2 , gdzie $n_2 = 1, 8, 32, 64, 128, 256, 1024$. Użytkownik ma do dyspozycji trzy źródła impulsów PCK2:

- generator sygnału zegarowego dla mikrosterownika,
- dedykowany generator impulsów, do którego wyprowadzeń TOSC1 i TOSC2 przyłącza się rezonator kwarcowy o częstotliwości 32768 Hz,
- zewnętrzne impulsy zegarowe podawanych na wejście TOSC1. Maksymalna częstotliwość tych impulsów wynosi

$$f_{\max} = \begin{cases} 256 \text{ kHz} & \text{dla } f_{CK} \geq 1 \text{ MHz} \\ 1/4 * f_{CK} & \text{dla } f_{CK} < 1 \text{ MHz} \end{cases}$$

gdzie f_{CK} - j.w.

Licznik T/C2 może zostać zaprogramowany w trybie odmierzenia czasu, porównywania oraz modulacji szerokości impulsów. Istnienie wewnętrznego generatora 32768 Hz ułatwia też realizację zegara czasu rzeczywistego.

Układ nadzorczy stanowi licznik taktowany impulsami o częstotliwości f_{OSC}/n_3 , gdzie f_{OSC} jest częstotliwością dedykowanego generatora, a współczynnik podziału $n_3 = 2^i * 10^3$, $i = 4, 5, \dots, 11$. Nominalna częstotliwość generatora wynosi 1 MHz przy napięciu zasilania mikrosterownika równym 5V.

Komparator analogowy dokonuje porównania napięć przyłożonych do wejść AIN0 i AIN1 mikrosterownika. Wynik porównania (1, jeśli $V_{AIN0} > V_{AIN1}$) jest dostępny do odczytu w rejestrze sterującym komparatora. Komparator może zgłaszać żądania przerwania, jego wyjście może też zostać przyłączone do wejścia przechwytyjącego licznika T/C1.

Interfejs SPI umożliwia zmianę zawartości wewnętrznej pamięci nieulotnej we wszystkich mikrosterownikach serii AT90S. W niektórych jest on dostępny także w trybie normalnej pracy dla realizacji komunikacji z otoczeniem. Mikrosterownik może być wówczas skonfigurowany jako układ nadrzędny lub podrzędny. Prędkość transmisji w interfejsie SPI jest równa, co do wartości liczbowej, częstotliwości impulsów zegarowych SCK. W mikrosterownikach omawianej serii częstotliwość tych impulsów stanowi $1/n_4$ ($n_4 = 4, 16, 64, 128$) częstotliwości zegarowej f_{CK} .

Przetwarzanie analogowo-cyfrowe jest realizowane metodą kompensacji wagowej. Rozdzielczość przetwornika wynosi 10 bitów, dokładność bezwzględna ± 2 LSB, a nieliniowość całkowita 0.5 LSB. Przetwornik wymaga dołączenia zewnętrznego: napięcia zasilania AV_{CC} , masy analogowej AGND oraz napięcia odniesienia AV_{REF}

(napięcie AV_{CC} nie może różnić się od napięcia zasilania mikrosterownika więcej niż $\pm 0.3V$, natomiast napięcie odniesienia musi mieścić się w przedziale $2V \pm AV_{CC}$). Wymaga też taktowania impulsami zegarowymi o częstotliwości z zakresu $50 \div 200$ kHz; częstotliwość taktowania ustala się poprzez odpowiedni dobór współczynnika n_5 podziału częstotliwości zegarowej f_{CK} ($n_5 = 2^j$, $j = 1, 2, \dots, 7$). Czas przetwarzania wynosi 13 taktów zegara, istnieje też możliwość jego wydłużenia do 25 taktów. Przetwornik może pracować w trybie:

- pojedynczego przetwarzania w wybranym kanale analogowym (*single conversion mode*),
- stale powtarzanej pętli przetwarzań wszystkich kanałów (*free running mode*).

Obciążalność linii równoległych portów wejścia/wyjścia we wszystkich mikrosterownikach serii AT90S wynosi 20 mA. Pozwala to na bezpośrednie z nich sterowanie diodami i wyświetlaczami LED. Ponadto mikrosterowniki AT90S1200 i AT90S2313 mają taki sam rozkład wyprowadzeń jak oferowane przez tego samego producenta mikrosterowniki serii AT89Cx051.

Mikrosterownik programowany szeregowo powinien być taktowany impulsami zegarowymi o częstotliwości mieszczącej się w określonym dla danego typu zakresie. Dla każdego typu mikrosterownika jest również określony czas trwania poszczególnych stanów logicznych impulsu SCK w funkcji częstotliwości zegarowej. Programowanie jest realizowane za pomocą 4-bajtowych rozkazów. Tabela 4 zawiera zestawienie ośmiu podstawowych rozkazów; listę rozkazów w niektórych mikrosterownikach rozszerzono o kilka pozycji. Sposób przesyłania rozkazów po liniach interfejsu SPI jest taki sam jak w przypadku mikrosterownika AT89S8252.

Tabela 4. Podstawowe rozkazy programowania szeregowego mikrosterowników serii AT90S

Rozkaz	Format rozkazu			
	bajt 1	bajt 2	bajt 3	bajt 4
rozpoczęcie programowania	10101100	01010011	xxxxxxx	xxxxxxx
kasowanie całej zawartości pamięci FLASH oraz EEPROM	10101100	100xxxx	xxxxxxx	xxxxxxx
odczyt wybranego bajtu słowa pamięci FLASH	0010H000	0000a ₁₁ ...a ₈	a ₇ a ₆ ...a ₁ a ₀	d ₇ d ₆ ...d ₁ d ₀
zapis wybranego bajtu słowa pamięci FLASH	0100H000	0000a ₁₁ ...a ₈	a ₇ a ₆ ...a ₁ a ₀	d ₇ d ₆ ...d ₁ d ₀
odczyt bajtu pamięci EEPROM	10100000	0000000a ₀	a ₇ a ₆ ...a ₁ a ₀	d ₇ d ₆ ...d ₁ d ₀
zapis bajtu pamięci EEPROM	11000000	0000000a ₀	a ₇ a ₆ ...a ₁ a ₀	d ₇ d ₆ ...d ₁ d ₀
programowanie bitów zabezpieczających	10101100	111xxlb ₂ lb ₁ x	xxxxxxx	xxxxxxx
odczyt wybranego bajtu sygnatury	00110000	xxxxxxx	xxxxxs ₁ s ₀	d ₇ d ₆ ...d ₁ d ₀
a _i – bity adresowe pamięci d _i – bity danych x – wartość bitu nieistotna	s _i – bity adresowe bajtów sygnatury H – wybór bajtu w słowie (0 - młodszy, 1 - starszy) lb _i – bity zabezpieczające (programowane, jeśli lb _i = 0)			

Uaktywnienie trybu programowania szeregowego następuje poprzez ustawienie stanu niskiego na linii zerowania mikrosterownika i wysłanie rozkazu rozpoczęcia programowania. Zmiana zawartości pamięci programu wymaga uprzedniego jej

wykasowania, natomiast zapisywany bajt pamięci EEPROM jest kasowany automatycznie. Minimalny czas zapisu bajtu pamięci nieulotnej wynosi 4 ms przy zasilaniu mikrosterownika napięciem 5V. Istnieje również możliwość uzyskania informacji o zaawansowaniu tej operacji. Operacja kasowania trwa, w zależności od typu mikrosterownika, 8÷10 ms. Ustawienie stanu wysokiego na linii zerowania kończy programowanie, po czym mikrosterownik rozpoczyna realizację programu od adresu 000H.

Program ładujący dla mikrosterowników serii AT90S jest dostępny bezpłatnie na stronie internetowej firmy AEC Electronics (<http://www.aec-electronics.co.nz>). Program ten wykorzystuje port drukarki mikrokomputera IBM PC; na wymienionej stronie znajdują się również zalecenia dotyczące elektrycznego połączenia mikrosterownika i mikrokomputera programującego.

4. Podsumowanie

Zastosowanie mikrosterowników z wewnętrzną pamięcią programu upraszcza strukturę części sprzętowej systemu docelowego. Dzięki temu zmniejszeniu ulegają: liczba połączeń w systemie, rozmiary obwodów drukowanych, moc pobierana przez system. Dodatkową korzyścią jest podniesienie niezawodności systemu.

Mikrosterownik z pamięcią programu typu FLASH jest tańszy od swojego odpowiednika zawierającego wielokrotnie programowalną pamięć EPROM. Niektóre takie mikrosterowniki mogą być programowane nie tylko za pomocą specjalistycznego programatora, ale również za pośrednictwem interfejsu szeregowego. Do realizacji programowania szeregowego w większości przypadków wystarcza tylko mikrokomputer, ponadto może ono zostać przeprowadzone nawet wtedy, gdy mikrosterownik znajduje się już w systemie docelowym. Upraszcza to proces uruchamiania systemu jak też wymianę oprogramowania w razie jego rozbudowy.

Przedstawione w niniejszej publikacji mikrosterowniki są dostępne w naszym kraju, a ich cena nie jest wygórowana. Mikrosterowniki te stanowią ciekawą alternatywę dla dotychczas stosowanych układów, a możliwość ich zaprogramowania bez konieczności posiadania specjalistycznych urządzeń może mieć dla niektórych osób istotne znaczenie.

Literatura

- [1] Anderson P. H.: Use of a PC Printer Port for Control and Data Acquisition. <http://et.nmsu.edu/~ctti/fall96/computer/printer/printer.html>
- [2] Caban D.: Synchroniczny interfejs szeregowy SPI i jego programowa realizacja. *Elektronizacja*, 11/2000.
- [3] Materiały informacyjne firmy Atmel Corp. <http://www.atmel.com/>.
- [4] Metzger P.: *Anatomia PC*. Helion, Gliwice 1993.
- [5] Motorola: *CMOS Logic Data*. DL131 Rev. 1., 1988.
- [6] Starecki T.: *Mikrokontrolery jednoukładowe rodziny 51*. NOZOMI, Warszawa 1996.

BIBLIOTEKA GŁÓWNA
Politechniki Śląskiej

P.3057/00

Druk: Drukarnia Gliwice, ul. Zwycięstwa 27, tel. 230 49 50