

Stanisław CIEŚLA, Arkadiusz TWARDON
Politechnika Śląska, Instytut Informatyki

AKWIZYCJA I PREZENTACJA DANYCH Z WYKORZYSTANIEM TECHNOLOGII INTERNETU I INTRANETU DLA POTRZEB KIOSKU INFORMACYJNEGO

Streszczenie. Artykuł przedstawia koncepcję wykorzystania technologii Internet i intranet do stworzenia kiosku informacyjnego, omawia ideę kiosku informacyjnego, przybliża technologię dynamicznego HTML'u, Obiektowy Model Dokumentu HTML, użycie technologii Active Server Pages oraz JavaScript do zrealizowania dostępu do bazy danych i interfejsu użytkownika.

USING INTERNET/INTRANET TECHNOLOGIES FOR DATA COLLECTING AND PRESENTING IN INFORMATION KIOSK

Summary. This article presents the idea of using the Internet/Intranet technologies to create an information kiosk. The following terms are discussed below: information kiosk, Dynamic HTML technology and Dynamic Object Model. We also described the solution of database access and user's interface creating problems using Active Server Pages technology and JavaScript language.

1. Koncepcja kiosku informacyjnego

Jako kiosk informacyjny możemy rozumieć platformę sprzętowo-programową umożliwiającą udostępnianie pewnych informacji użytkownikowi. Informacje te mogą być przekazywane na różne sposoby i z użyciem różnorodnych mediów przekazu. Informacja przekazywana powinna być tak zorganizowana, by użytkownik mógł w sposób prosty i szybki osiągnąć interesujący go poziom wiedzy.

Kiosk informacyjny może mieć bardzo szerokie spektrum zastosowań, począwszy od prostego kilkupoziomowego menu wzbogaconego graficznie informującego o rozmieszczeniu pomieszczeń w zespole budynków, w których kiosk się znajduje, lub informującego o usłu-

gach świadczonych przez właściciela kiosku, do zaawansowanych zastosowań obejmujących wizualizowanie procesów, interakcję z użytkownikiem celem określenia jego potrzeb, a następnie dostarczenia mu spodziewanej informacji.

2. Kategorie kiosków

Na podstawie powyższych rozważań oraz dokonanych obserwacji wyodrębniono trzy podstawowe kategorie kiosków informacyjnych ze względu na swoją koncepcję udostępniania informacji.

- Kategoria pierwsza obejmuje jedynie dostarczanie informacji, prostą nawigację pomiędzy przekazywanymi porcjami wiedzy.
- Kategoria druga rozszerza zakres usług wymieniony w pierwszej kategorii o pobieranie pewnych informacji od użytkownika, czy to na sposób wyboru w prostszej formie, czy też w sposób bardziej skomplikowany pobierając od użytkownika dane w postaci tekstów lub dokumentów.
- Kategoria trzecia przez analogię rozszerza funkcjonalność poprzedniej kategorii o dodatkową logikę oraz pewne działania uzależnione od dostarczonej przez użytkownika informacji, a wpływające na informacje użytkownikowi zwracane.

Podane powyżej kategorie nie mają nic wspólnego z platformą programowo–sprzętową realizacji i służą jedynie do sklasyfikowania obszarów zastosowań i wyodrębnienia pewnych poziomów świadczenia usług.

3. Koncepcja środowiska programowo–sprzętowego realizująca ideę kiosku informacyjnego

Jako platformę badań nad implementacją idei dyskutowanej w opracowaniu wybrano technologię i mechanizmy Internetu/intranetu. Jako system klienta, za pomocą którego „kiosk” prezentowałby informację, wybrano przeglądarkę WWW, jako „protokół” transmisji zaproponowano protokół HTTP stosowany dla usług WWW, natomiast jako źródło informacji oraz mechanizm realizujący dostarczanie informacji, innymi słowy zasilanie „kiosku”, posłużył serwer WWW.

Jako języka implementacji przekazywanej informacji użyto języka HTML w swym ostatnim rozszerzeniu określanego mianem dynamicznego lub oficjalnie wersji 4.0 tegoż języka.

Wybór platformy WWW jako środowiska dla implementacji został podyktowany praktycznie pełną funkcjonalnością technologii WWW w zakresie wspomnianej wyżej pierwszej

kategorii kiosku informacyjnego. Technologia WWW w swym założeniu miała służyć do przekazywania informacji graficzno–tekstowej, powiązanej ze sobą łąkami logicznymi. W swym procesie ewolucji została rozszerzona o dodatkowe możliwości przesyłania mediów innych niż tylko grafika i tekst. W ostatnim czasie gwałtowny rozwój technologii Internetu i intranetu spowodował rozszerzenie możliwości języka i ogólnie pojętej technologii WWW o elementy pewnych akcji i działań w procesie przesyłu informacji do użytkownika; chodzi tu o skryptowe języki programowania (JavaScript i VBScript) oraz obiekty włączane do przesyłanej informacji, a napisane w innych językach programowania, takie jak komponenty ActiveX i Java Applets.

Rozszerzono także możliwości serwerów WWW, które obecnie nie tylko potrafią dostarczać strony z informacją, ale także strony te dynamicznie tworzyć w oparciu o pewne algorytmy i opracowany kod, jak również współpracować z innymi usługami systemów operacyjnych, włączając je do środków dostępnych twórcom skomplikowanych serwisów WWW. Można tu mówić o połączeniu z bazami danych czy też z innymi aplikacjami napisanymi specjalnie na potrzeby jakiegoś konkretnego rozwiązania.

Taki zakres dostępnych środków pozwala na wykorzystanie platformy WWW także dla drugiej i trzeciej kategorii kiosku informacyjnego.

Wybór technologii WWW ma także inne korzyści, które zostaną poniżej przybliżone:

- Jest to rozwiązanie o charakterze multimedialnym, przeglądarki są przygotowane do prezentowania dźwięku, sekwencji video oraz innych przesyłanych do nich dokumentów.
- Platforma ma charakter niezależny programowo i sprzętowo; jeżeli zostanie przyjęty pewien standardowy poziom implementacji wykorzystywany przez wszystkich producentów przeglądarek WWW i zrezygnuje się ze specyficznych rozszerzeń, można przyjąć, iż prezentowana usługa jest dostępna dowolnemu klientowi.
- Podobnie ma się niezależność platformy systemu operacyjnego; jeżeli przyjąć, iż rozwój technologii Internetu jest ściśle związany z rozwojem systemów operacyjnych, to mamy gwarancję, iż opracowane rozwiązanie będzie działało na kolejnych wersjach systemu operacyjnego bez konieczności jego całkowitego przebudowania.
- Przyjęcie technologii WWW pozwala na dużą skalowalność stosowanego rozwiązania; nie ma znaczenia, czy rozwiązanie ogranicza się do pojedynczego komputera, czy też działa w sieci lokalnej czy globalnej, w każdym z tych przypadków kod i zastosowana implementacja są jednakowe i nie wymagają przeróbek, by stosować je w szerszym czy węższym zakresie.
- Niebagatelnym aspektem jest tutaj także cena i czas tworzenia aplikacji; wykorzystując elementy technologii WWW, język HTML, poruszamy się jakby ponad językami wysokiego poziomu, korzystając z pewnych gotowych elementów tworzących roz-

wiązanie; również rozwój aplikacji, jej modyfikacje są dokonywane w jednym miejscu i natychmiast dostępne u wszystkich odbiorców.

4. Przykładowa realizacja aplikacji typu kiosk informacyjny

Jako przykład wdrożenia idei dyskutowanej w opracowaniu stworzono stanowisko laboratoryjne służące jako platforma do ćwiczeń w zakresie przedmiotu Aplikacje WWW prowadzonego w Zakładzie Teorii i Projektowania Systemów Komputerowych Instytutu Informatyki.

Do zrealizowania przykładowego wdrożenia użyto technologii Active Server Pages po stronie serwera do realizacji operacji na bazie danych oraz DHTML i JavaScript po stronie klienta do realizacji interfejsu użytkownika.

Poniżej przedstawiono i scharakteryzowano wybrane zagadnienia użyte do realizacji wdrożenia.

4.1. Technologia Active Server Pages

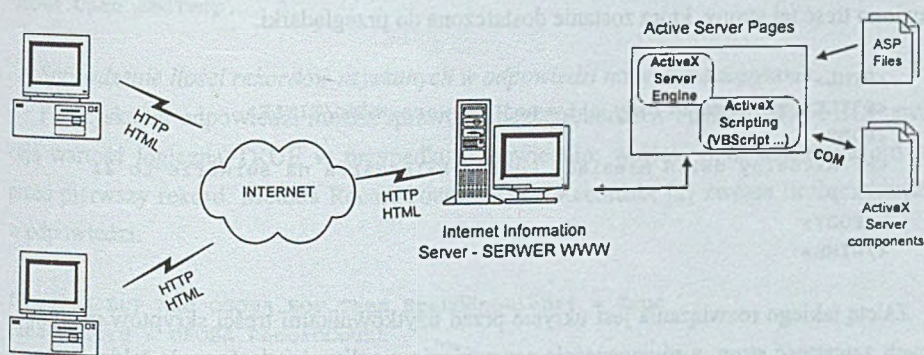
Technologia Aktywnych Stron Serwera (ang. Active Server Pages) została wprowadzona przez firmę Microsoft jako rozszerzenie kolejnej wersji serwera WWW oferowanego przez tę firmę pod nazwą Internet Information Server. Technologia ASP umożliwia zmianę wyglądu i treści stron WWW na serwerze przed ich przesłaniem do przeglądarki komputera użytkownika w odpowiedzi na jego żądanie dostępu do tej strony. Zmiana treści i wyglądu strony WWW jest możliwa dzięki umieszczeniu w jej treści tekstu programu w formie skryptu [1].

Mechanizm ASP został wyposażony w możliwość rozszerzania funkcjonalnego zakresu realizowanych operacji poprzez wykorzystywanie dodatkowo instalowanych komponentów ActiveX. W ramach standardowej instalacji dostarczany jest komponent ADO pozwalający na dostęp do baz danych poprzez interfejs ODBC. W przypadku gdy dostarczony przez producenta zestaw komponentów nie zaspokaja w pełni wymagań dotyczących funkcjonalności rozwiązania, nie stoi na przeszkodzie w doinstalowaniu i zarejestrowaniu dodatkowego komponentu stworzonego specjalnie na potrzeby konkretnej realizacji bądź zakupionego u innego dostawcy oprogramowania.

Skrypt z instrukcjami, które mają być wykonane, może być umieszczony pomiędzy instrukcjami języka HTML, w miejscu w którym treść strony ma zostać zmieniona i jest zależna od rezultatu wykonania pewnych instrukcji. Jest to technologia podobna do znanych wcześniej rozwiązań pozwalających na umieszczanie w treści strony napisanej w języku HTML fragmentów napisanych w innych językach (np. w językach skryptowych VBScript

czy JavaScript), które mogą być wykonane w przeglądarce użytkownika po załadowaniu danej strony.

Wadą takiego rozwiązania jest duża zależność tworzonych stron zawierających skrypty od typów przeglądarek, które mają te strony interpretować. Nie wszystkie przeglądarki potrafią wykonywać skrypty, nie wszystkie też wykonują je w ten sam sposób. Taka metoda modyfikowania stron nie jest wygodna, a poza tym treść skryptów jest wprost udostępniona użytkownikom, co w niektórych sytuacjach może być niewygodne, a czasami nawet niebezpieczne.



Rys. 1. Schemat działania technologii Active Server Pages
Fig. 1. Short description of how Active Server Pages works

Technologia ASP pozwala na wykorzystanie mechanizmu skryptów na stronach WWW, ale pozwala na ich wykonywanie w środowisku serwera WWW i zapewnia, że do przeglądarki na komputerze użytkownika dociera odpowiednio zmodyfikowana już treść strony w języku HTML. Technologia ASP pozwala nie tylko na umieszczanie w treści stron kodu w językach skryptowych, ale także na wywołanie z wnętrza stron WWW zewnętrznych komponentów ActiveX, działających w standardzie COM [1].

Schemat przepływu stron WWW, z użyciem technologii ASP, między przeglądarką użytkownika a serwerem WWW został pokazany na rys. 1.

Poniżej przedstawiono przykładową treść strony wykorzystującą technologię ASP [2].

```
<HTML>
<TITLE>Przykład strony aktywnej serwera WWW</TITLE>
<BODY>
<H2>Aktualny dzień miesiąca według kalendarza na serwerze to
<% Call PrintDate %>
</H2>
<SCRIPT LANGUAGE=Jscript RUNAT=Server>
function PrintDate()
```

```

    {
        var x
        x = new Date()
        Response.Write(x.getDate())
    }
</SCRIPT>
</BODY>
</HTML>

```

Powyższa strona została napisana z użyciem języka skryptowego JavaScript i powoduje odczyt daty serwera i umieszczenie jej w treści przesyłanej do użytkownika. Poniżej przedstawiono treść tej strony, która zostanie dostarczona do przeglądarki.

```

<HTML>
<TITLE>Przykład strony aktywnej serwera WWW</TITLE>
<BODY>
<H2>Aktualny dzień miesiąca według kalendarza na serwerze to 21
</H2>
</BODY>
</HTML>

```

Zaletą takiego rozwiązania jest ukrycie przed użytkownikami treści skryptów modyfikujących zawartość stron, a równocześnie zapewnienie możliwości dostępu do takich stron za pomocą dowolnej przeglądarki WWW.

4.2. ASP – tworzenie i manipulacje na danych

Dostęp do baz danych realizowany jest poprzez wykorzystanie obiektów i metod dostarczonych przez komponent ActiveX ADO. Pozwala on na wykonanie praktycznie dowolnych manipulacji na danych. Poniżej przedstawiamy przykłady implementacji najczęściej wykonywanych operacji.

Utworzenie połączenia z bazą danych

Poniższy przykład ilustruje zasadę tworzenia połączenia z bazą danych. Parametry DC_x określają odpowiednie parametry połączenia. Obiekt Session [2] zawiera zestaw zmiennych zdefiniowanych przez programistę, a przechowujących wartości charakterystyczne dla konkretnej sesji przeglądarka-ośrodek WWW.

```

Set DC = Server.CreateObject("ADODB.Connection")
DC.ConnectionTimeout = Session("DC_ConnectionTimeout")
DC.CommandTimeout = Session("DC_CommandTimeout")
DC.Open Session("DC_ConnectionString"), Session("DC_RuntimeUserName"),
Session("DC_RuntimePassword")

```

Utworzenie struktury danych zawierającej wynik zapytania

Po utworzeniu połączenia z bazą danych możliwe jest wykonanie zapytania SQL. W odpowiedzi DBMS zwraca zestaw rekordów będący odpowiedzią na zadane pytanie. Zestaw ten nazywany jest "recordset" i wykorzystywany do manipulowania na danych.

```
Set cmdTemp = Server.CreateObject("ADODB.Command")
Set OSOBA = Server.CreateObject("ADODB.Recordset")
cmdTemp.CommandText = "SELECT id, imie, nazwisko FROM prac"
cmdTemp.CommandType = 1
Set cmdTemp.ActiveConnection = DC
OSOBA.Open cmdTemp, , 1, 3
```

Sprawdzenie ilości rekordów uzyskanych w odpowiedzi na wydane zapytanie

Po uzyskaniu odpowiedzi można sprawdzić liczbę rekordów. Funkcje EOF i BOF zwracają wartość logiczną TRUE w przypadku odpowiednio: wyjścia poza ostatni rekord lub przed pierwszy rekord. Metoda RecordCount obiektu Recordset [2] zwraca liczbę rekordów w odpowiedzi.

```
If OSOBA.EOF And OSOBA.EOF Then BrakOdpowiedzi = True
IloscRekord = OSOBA.RecordCount
```

Poruszanie się w obrębie rekordów odpowiedzi

Do poruszania się o w obrębie rekordów odpowiedzi służą odpowiednie metody obiektu Recordset:

- przejście do następnego rekordu – MoveNext;
- przejście do poprzedniego rekordu – MovePrev;
- przejście do pierwszego rekordu – MoveFirst;
- przejście do rekordu ostatniego – MoveLast.

Dodawanie nowego rekordu

Dodawanie nowego rekordu jest możliwe na dwa sposoby. Po pierwsze można wykonać zapytanie SQL typu INSERT. Po drugie można skorzystać z metody AddNew i Update obiektu Recordset do utworzenia nowego rekordu i zapisania jego treści do bazy danych. Tą drugą metodę ilustruje poniższy przykład:

```
Set TEL = Server.CreateObject("ADODB.Recordset")
TEL.CursorType = 1
TEL.LockType = 3
TEL.Open "telefony", DC, , , 2
TEL.AddNew
TEL("id") = Session("id")
```

```

TEL("tel") = Request.Form("tel")
TEL("data") = Request.Form("data")
TEL("rodzaj") = Request.Form("rodzaj")
TEL("opis") = Request.Form("opis")
TEL.Update
TEL.Close

```

Usuwanie rekordu istniejącego

Usuwanie rekordu istniejącego także jest możliwe na dwa sposoby. Po pierwsze, można wykonać zapytanie SQL typu DELTE. Po drugie, można skorzystać z metody Delete obiektu Recordset do usunięcia rekordu z bazy danych. Przedstawiony przykład ilustruje sposób usuwania rekordu, dla którego wartość pola id wynosi 5.

```

TEL.Open "SELECT tel, data FROM TELEFONY WHERE id=5", DC, , , 2
TEL.Delete
TEL.Close

```

Zmiana treści rekordu

Podobnie jak w poprzednich wypadkach zawartość rekordu można zmodyfikować poprzez zapytanie SQL typu UPDATE lub korzystając z metody Update obiektu Recordset. Poniższy przykład pokazuje wykorzystanie metody Update obiektu Recordset.

```

TEL.Open "SELECT tel, data FROM TELEFONY WHERE id=5", DC, , , 2
TEL("tel") = Request.Form("tel")
TEL("data") = Request.Form("data")
TEL.Update
TEL.Close

```

Odświeżenie zawartości zbioru rekordów

Odświeżenie zawartości zbioru rekordów możliwe jest poprzez użycie metody Requery obiektu Recordset.

```

SqlStr = "SELECT tel, data FROM TELEFONY WHERE id=5"
TEL.Open SqlStr, DC, , , 2
'tutaj pojawia sie dalszy kod programu
SqlStr = "SELECT tel, data FROM TELEFONY WHERE id=10"
TEL.Requery
'rekordset zawiera nowy zestaw rekordów

```

4.3. Technologia Document Object Model i język JavaScript

Głównym celem wprowadzenia języka JavaScript [3] do środowiska przeglądarek internetowych jest rozszerzenie możliwości stron WWW o element interakcji z użytkownikiem

iw ten sposób, by procedury programowe umieszczone na stronie wykonywały się w przeglądarce bez konieczności odwołania się do serwera WWW i ponownego dostarczenia zmodyfikowanej strony do przeglądarki.

Mówiąc o interakcji na stronie, mamy na myśli śledzenie pewnych działań użytkownika i reagowanie na nie modyfikacją zawartości prezentowanej strony. Jako sposoby interakcji z użytkownikiem wykorzystuje się cały zasób języka HTML odnoszący się do różnego rodzaju elementów dialogowych, to jest: klawiszy, list itd.

Aby umożliwić operowanie elementami języka HTML umieszczonymi na stronie, z poziomu języka skryptowego zaimplementowano na poziomie przeglądarki definicje pewnych elementów języka HTML oraz struktury dokumentu HTML'owego w postaci obiektów zebranych w hierarchię. Ten uporządkowany zbiór obiektów nosi nazwę Document Object Model [5] i został zaproponowany przez organizację W3C, oraz z pewnymi modyfikacjami przyjęty i zaimplementowany przez wiodących producentów przeglądarek.

Obiekty dostępne z poziomu języka posiadają zdefiniowane zbiory właściwości, które reprezentują stan obiektu oraz zbiór metod przewidzianych dla danego obiektu odwzorowujący dostępne na danym obiekcie operacje.

Wreszcie większość obiektów posiada możliwość zdefiniowania i określenia obsługi pewnych zdarzeń, jakie mogą na danym obiekcie zajść wywołane działaniami użytkownika.

Korzystając więc w treści skryptów z obiektów, ich właściwości, metod i zdarzeń, można z poziomu języka programowania dowolnie kreować wygląd strony, dynamicznie ją modyfikować w zależności od reakcji użytkownika bez jakiegokolwiek konieczności odwołania się do serwera WWW.

Odwołania do serwera zostają ograniczone do żądań dostarczenia kolejnej porcji danych lub przejścia pod inny adres.

Poniżej przedstawiono i skomentowano kilka wybranych przykładów użycia w skryptach obiektów, ich metod i właściwości.

4.3.1. Detekcja typu przeglądarki

Wykrycie typu używanej przeglądarki możliwe jest przez wykorzystanie Obiektowego Modelu Dokumentu będącego podstawą do tworzenia stron w Dynamicznym HTML. Informację o typie i wersji używanej przeglądarki przechowuje obiekt navigator.

Poniżej przedstawiono przykładowy kod funkcji sprawdzającej typ przeglądarki [4]. Funkcja napisana jest w języku JavaScript i jako element strony przetwarzana jest po stronie klienta.

```
<script language="JavaScript">
```

```
<!--
```

```

function detect()
{
    if (navigator.appName == "Netscape" && parseInt (navigator.appVersion)
    >= 3)
    {
        // akcje wykonywane w przypadku wykrycia Netscape Navigator'a
    }
    else
    if (navigator.appName == "Microsoft Internet Explorer"
        && parseInt (navigator.appVersion) >= 4)
    {
        // akcje wykonywane w przypadku wykrycia IE 4.x
    }
    else
    {
        // akcje wykonywane w przypadku wykrycia innej przeglądarki
    }
}
-->
</script>

```

4.3.2. Otwieranie kolejnej wersji przeglądarki

Otwieranie nowego okna przeglądarki umożliwia metoda `open()` obiektu `window`. Poniżej przedstawiono dwa wywołania tej funkcji. Pierwsze z nich otwiera nową wersję Netscape Nawigatora i czyni ją aktywną, drugie otwiera nową wersję IE. W przypadku IE nowo otwarte okno automatycznie staje się aktywne.

```

menuMain = window.open (url, "menu", "fullscreen=1, toolbar=0, location=0,
directories=0, status=1, menubar=0, scrollbars=0, resizable=0, width=790,
height=600");
menuMain.focus();

```

```

menuMain = window.open (url, "menu", "fullscreen=1, toolbar=0, location=0,
directories=0, status=0, menubar=0, scrollbars=0, resizable=0, width=800,
height=600, top=0, left=0");

```

Znaczenie parametrów wywołania metody `window.open()`

Pierwszym argumentem jest adres strony, która zostanie załadowana do nowo otwartego okna przeglądarki. Drugim argumentem wywołania jest tytuł okna, natomiast w trzecim określamy poszczególne parametry otwarcia:

`Channelmode` - pokazuje / ukrywa listę kanałów CDF

Wartości:

0 - niewidoczna

1 - widoczna

Directories – Pokazuje / ukrywa przycisk katalogów

Wartości:

0 - niewidoczny

1 - widoczny

Fullscreen – Określa, czy okno ma być rozciągnięte na całym ekranie

Wartości:

0 - standardowy rozmiar

1 - maksymalny rozmiar

Location - Pokazuje / ukrywa pasek adresowy

Wartości:

0 - niewidoczny

1 - widoczny

Menubar - Pokazuje / ukrywa menu przeglądarki

Wartości:

0 - niewidoczna

1 - widoczna

Status – Pokazuje / ukrywa status okna

Wartości:

0 - niewidoczny

1 - widoczny

Resizable – Określa, czy okno może zmieniać rozmiar

Wartości:

0 - rozmiar stały

1 - rozmiar zmienny

Scrollbars - Pokazuje / ukrywa paski przewijania

Wartości:

0 - niewidoczne

1 - widoczne

Toolbar – Pokazuje / ukrywa standardowy pasek narzędzi

Wartości:

0 - niewidoczny

1 - widoczny

Height - wysokość okna w pikselach

Wartość liczbową

Width - szerokość okna w pikselach

Wartość liczbową

Top - położenie względem góry ekranu

Wartość liczbowa

Left - położenie względem lewej strony ekranu

Wartość liczbowa

4.3.3. *Uaktywnienie elementów tabeli na akcje użytkownika*

Uaktywnienie elementów tabeli na działania użytkownika umożliwia mechanizm zdarzeń wchodzący w skład Dynamicznego HTML [2, 3, 4, 5]. Zdarzenie (np. przesunięcie myszki nad wybrany element) może zostać obsłużone przez procedurę napisaną w języku skryptowym. Obsługa może polegać na zmianie wyglądu czy treści elementu lub na wykonaniu określonej operacji, np. załadowaniu nowej strony. Poniżej przedstawiono przykład ilustrujący zasadę obsługi zdarzeń `onMouseOver` i `onMouseOut` polegającą na zmianie wyglądu komórki tabeli oraz zdarzenia `onMouseClick` polegającą na załadowaniu nowej strony.

W przykładzie wykorzystywane są definicje dwóch stylów określających kolor używanej czcionki.

```
<td class="nButton" onMouseOver="this.className='bButton' "
onMouseOut="this.className='nButton' "
onClick="location.href='zaj_list.asp'">Lista prowadzonych przedmiotów</td>
```

Inne zdarzenia, jakie mogą zostać obsłużone przez skrypty, to:

`OnBlur` - okno przestaje być aktywne

`OnFocus` - okno jest aktywne

`OnHelp` - użytkownik wciśnie F1 lub uruchomi Help z menu

`OnBeforeUnload` - okno przeglądarki zostaje zamknięte

`OnResize` - zmienia się rozmiar okna

`OnLoad` - dokument zostaje załadowany do pamięci

`OnUnload` - dokument zostanie wyładowany z pamięci

`OnScroll` - strona jest przewijana

`OnError` - wystąpił błąd ładowania obiektu, np. bitmapy.

LITERATURA

1. Cieśla S., Małysiak-Cieśla B.: Wykorzystanie technologii Active Server Pages w dostępie do baz danych w sieciach Internet i intranet, ZN Pol. Śl. Informatyka z. 34, Gliwice 1998.
2. Microsoft Developer Network, January 1999, Microsoft Press.

3. Dynamic HTML Technologies, JavaScript Technologies on [HTTP://WWW.NETSCAPE.COM](http://WWW.NETSCAPE.COM).
4. DHTML, HTML & CSS at Web WorkShop on [HTTP://MSDN.MICROSOFT.COM](http://MSDN.MICROSOFT.COM).
5. HTML, Style Sheet CSS, Document Object Model on [HTTP://WWW.W3C.ORG](http://WWW.W3C.ORG).

Recenzent: Dr inż. Leszek Płonka

Wpłynęło do Redakcji 8 kwietnia 1999 r.

Abstract

Information kiosk consist of two main parts. Hardware platform is the first one. The other one is the specialised software. Generally speaking these two parts together may present any kind of information, one may think of. Information can be presented as a sequence of slides or as the answers on user's questions.

In the first case information has to be properly organised. A user should achieve satisfying level of knowledge in quick and simple way.

In the second case the problem is defined by the questions: how to create a client side application with the proper user's interface and how to connect database on the server side?

As a solution of those problems we recommend using World Wide Web platform. A WWW browser is the client side application. User's interface is developed on Dynamic HTML connected with JavaScript language. Our recommendation for the server side is the Internet Information Server. It allows using Active Server Pages technology for data processing. Database access is made by Active Data Object ActiveX component – a part of standard IIS installation.