

Standardy w projektowaniu WWW

Marcin Oczkowski

Katolickie Centrum Edukacji Młodzieży „KANA”
w Katowicach
pl. ks. E. Szramka 4
oczy@kana.silesia.top.pl

1. Wstęp

Serwis WWW czy pojedynczą stronę projektuje się zazwyczaj biorąc pod uwagę jeden z celów, którymi mogą być:

- chęć dotarcia do jak największej liczby użytkowników Internetu
- stworzenie efektywnego serwisu
- prezentacja konkretnej informacji
- zaprezentowanie własnych umiejętności

Niektóre z powyższych celów pozostają w konflikcie z innymi, co wymaga od projektanta starannego ich wyważenia, co dotyczy szczególnie dwu pierwszych celów. W przeciwnym wypadku powstanie bądź strona zawierająca mnóstwo informacji, aczkolwiek przerażająco nudna i nieciekawa estetycznie, bądź efektem będzie coś, co można będzie, nie narażając się na napotkanie tylko i wyłącznie komunikatów o błędach, obejrzeć tylko z wykorzystaniem jednego konkretnego programu klienckiego, czyli mówiąc prosto: przeglądarki; nie wspominając już o stronach, na których zapanuje taki natłok efektów specjalnych, że na treść nie starczy miejsca, co z kolei zdarza się bardzo często w wypadku projektantów pragnących się pochwalić li tylko umiejętnościami w programowaniu, czy posiadaniem najnowszych narzędzi.

Tekst poniższy traktuje o tym, co jest niezbędne, by serwisy WWW docierały do swych odbiorców w postaci czytelnej, jak najbardziej zbliżonej do intencji projektanta i pozbawionej błędów — o standardach. Oprócz standardów zostaną przedstawione także podstawy działania protokołu HTTP i tworzenia skryptów CGI.

2. Rodzaje standardów

Można wyróżnić kilka rodzajów standardów, które należy uwzględnić przy projektowaniu i tworzeniu serwisów WWW. Pierwszym z nich są standardy oficjalne, zawierające specyfikację protokołów, specyfikacje języków używanych przy

programowaniu WWW, sposób kodowania znaków narodowych; drugim — standardy nieformalne, nie ustanawiające norm, będące jednak pomocą dla osób odpowiedzialnych za tworzenie stron i serwisów Web; można by do nich zaliczyć podział obowiązków w grupie *webmasterów*, język stron, wielkość zawartych plików graficznych itp. Standardy oficjalne są tymi, których przestrzegać należy **zawsze i bezwzględnie**, standardy nieformalne stanowią jedynie ułatwienie. Sposób, w jaki mogą być przestrzegane standardy jest różny — niekiedy opowiadzenie się za standardem ma miejsce w chwili wyboru oprogramowania (np. serwer http), w innych zaś przypadkach jest kwestią stosowania odpowiednich norm w pracy projektanta czy administratora.

Konkretne standardy, mające bezpośredni związek z projektowaniem i uruchamianiem serwisów WWW to:

1. Standardy oficjalne
 - standard definiujący protokół HTTP (najnowsza wersja 1.1)
 - standard definiujący język HTML (najnowsza wersja 4.0)
 - standard definiujący sposoby kodowania znaków narodowych
 - specyfikacje używanych języków programowania, w wypadku, gdy stroną, po której odbywa się wykonanie programu jest strona klienta (Java, JavaScript)
2. Standardy nieoficjalne
 - język używany na stronach serwisu
 - standardy bezpieczeństwa
 - wielkość pojedynczej strony, wielkość plików graficznych prezentowanych na stronach

2.1. Protokół HTTP

Protokół HTTP określa, najprościej mówiąc, metody porozumiewania się naszej przeglądarki z serwerem WWW abyśmy mogli obejrzeć wybraną stronę WWW. Zasada działania tego protokołu jest następująca: przeglądarka wysyła do serwera żądanie o wysłanie określonego dokumentu oraz kilka innych, istotnych dla serwera informacji, a serwer wysyła do przeglądarki dokument wraz z serią innych informacji. Każde zapytanie (żądanie) klienta (czyli przeglądarki) składa się z trzech części:

- linii zapytania (klient: polecenie *method*, adres dokumentu i wersja protokołu HTTP; serwer: wersja protokołu HTTP, kod stanu serwera, opisujący m.in. informację czy dana strona istnieje, czy nie i opis.
- listy nagłówków (klient: opcjonalne informacje powiadamiające serwer m.in. o tym, jakie rodzaje dokumentów potrafi obsłużyć przeglądarka, typ przeglądarki itp.; serwer: dane dotyczące serwera i dokumentu)

- części zasadniczej (klient: informacje dodatkowe wykorzystywane przez serwer np. podczas wykonywania skryptów CGI; serwer: treść dokumentu, wynik działania skryptu CGI bądź, w wypadku gdy żądany dokument nie istnieje, bądź skrypt nie może być wykonany, czytelną informację o zaistniałym błędzie)

Specyfikację protokołu HTTP w wersji 1.1 można znaleźć w dokumencie [9]. Jest ona istotna bezpośrednio tylko dla twórców serwerów i przeglądarek WWW, jednak znajomość mechanizmu działania HTTP przydatna jest także podczas projektowania stron (zob. przykład 1 w części „Przykłady rozwiązań praktycznych”

2.2. Język HTML

„Hypertext Markup Language (HTML) jest prostym językiem formatowania tekstu używanym do tworzenia dokumentów hipertekstowych niezależnych od platformy”[11]. To HTML jest językiem, który sprawia, iż oglądane przez nas strony wyglądają tak samo (a przynajmniej podobnie), niezależnie od tego, czy korzystamy z przeglądarki Mosaic, Arena, Internet Explorera czy Netscape Navigator. Najnowsza specyfikacja HTML opisuje jego wersję oznaczoną numerem 4.0, większa część przeglądarek WWW pozwala na pełne korzystanie jedynie ze stron napisanych w starszych wersjach języka. Zdecydowanie najbezpieczniej stosować jest wersję 2.0 HTML, aby być pewnym, że zaprojektowane przez nas strony będą się bez problemów oglądać na wszystkich przeglądarkach. Generalnie zawsze, a przynajmniej wtedy, jeśli kierujemy się tym, by ze strony korzystało jak najwięcej użytkowników, przy projektowaniu korzystać ze specyfikacji HTML o jedną wersję wstecz względem najnowszej (w chwili obecnej odpowiada to wersji 3.2). Pamiętać należy także o tym, że właściwie żaden z popularnych programów służących do korzystania z WWW nie jest w pełni zgodny z oficjalną specyfikacją HTML, co spowodowane jest powolnym, w stosunku do rozwoju wymagań użytkowników, rozwojem tejże specyfikacji. Aby sprostać wymaganiom i rosnącym możliwościom oferowanym przez Internet, firmy oferujące przeglądarki wprowadziły zmiany i rozszerzenia obejmujące rozwiązania niektórych, nie istniejących w standardzie, a oczekiwanych i potrzebnych funkcji, głównie, choć nie tylko, obejmujących funkcje multimedialne.

Z dużą ostrożnością należy traktować rozszerzenia stosowane przez poszczególnych producentów przeglądarek, nawet tych, które stosuje największa ilość Internautów. Nieufność tą uzasadnia przede wszystkim niemożność pewnego stwierdzenia, który z programów jest rzeczywiście najpopularniejszy (większość publikowanych sondaży jest częścią działań marketingowych producentów oprogramowania) oraz, co być może ważniejsze, nietrwałość takich standardów; któż bowiem zagwarantuje, że uznana za najpopularniejszą przeglądarka będzie taką za rok al-

bo dwa. Przeprojektowywanie strony od początku jest bardziej pracochłonne niż stosowanie standardu.

Należy także pamiętać o użytkownikach, korzystających z przeglądarek tekstowych (wbrew pozorom np. lynx jest bardzo popularną przeglądarką). Jeśli nie jest możliwe stworzenie osobnej wersji strony, przeznaczonej dla przeglądarek tekstowych, dobrze jest pamiętać przynajmniej o umieszczeniu opisów do rysunków, które będą się pojawiać zamiast nich (przydatne także dla osób, które mają, ze względu na wolną transmisję — np. połączenie modemowe — wyłączoną obsługę rysunków w przeglądarce graficznej).

2.3. Kodowanie znaków narodowych

Czas przejść do problemu, który od dawien dawna stanowi powód wielu tzw. *flame wars*¹. Jest nim problem kodowania polskich znaków w serwisach WWW. Wydawałoby się, iż problem nie istnieje: Polska Norma (PN-93 T-42118) definiuje bowiem odpowiedni sposób kodowania. Polska Norma ma jednak to do siebie, iż obowiązuje jedynie w Polsce, natomiast większość oprogramowania powstaje jeśli nie za oceanem, to przynajmniej na pewno za granicą. Sam ten fakt nie powodowałby jeszcze zamieszania, biorąc pod uwagę, że Polska Norma zgodna jest w tym względzie ze standardem ogólnosiwiatowym, ISO-8859-2. Niestety, niektórzy producenci oprogramowania usiłowali wypromować własne pomysły dotyczące tego, jak mają być reprezentowane znaki polskiego alfabetu. Mowa o firmie Microsoft, która wprowadziła tzw. „stronę kodową 1250”, usiłując metodą „faktów dokonanych” uczynić ją standardem. Internet jednak, w przeciwieństwie do rynku aplikacji biurowych nie został praktycznie zmonopolizowany przez „giganta z Redmond”, Microsoft z czasem uznał ISO-8859-2 za standard w sieci i wszystkie aplikacje internetowe tej firmy pozwalają na jego obsługę. Problem nie zniknął jednak — w większości programów działających na platformie MS Windows CP-1250 nadal obowiązuje. Dotyczy to również narzędzi do tworzenia stron WWW, a także powoduje trudności przy konwertowaniu dokumentów napisanych przy pomocy innych aplikacji. Gdzieniedzie w Internecie nadal pokutują strony, na których polskie znaki są do odczytania jedynie dla osób używających MS Windows, a co gorsza zdarza się nadal (choć na szczęście zauważalna jest tendencja spadkowa) nowe strony kodowane w CP-1250. Zupełnie niezrozumiałe jest (i wyraźnie nacechowane złą wolą) promowanie CP-1250 w publikacjach mających na celu nauczyć podstaw pisania stron w języku HTML i ze swej strony stanowczo odradzam korzystanie z takowych „samouczków”.

Niestety problem z kodowaniem polskich znaków nie kończy się na konflikcie ISO-8859-2 z CP-1250. Oliwy do ognia dolewa standard języka HTML, proponu-

¹Czyli takie dyskusje, w których głównym celem jest zdenerwowanie przeciwnika

jąc własne metody oznaczania znaków nie znajdujących się w alfabecie łacińskim, z których korzystają niektóre aplikacje służące do tworzenia stron WWW, co gorsza niezależnie od platformy systemowej (te same metody stosuje np. FrontPage Microsoftu, jak i tryb PGSGML w uniksowym edytorze Emacs). Generalnie, problem ten dotyczy narzędzi, które przewidują tworzenie stron zgodnych z ISO-8859-1, jednak dobrze jest zwrócić uwagę (podglądając efekt działania programu jakimkolwiek edytorem tekstowym), czy zamiast znaków polskiego alfabetu nie pojawiają się znaczki typu „´”.

2.4. Języki programowania WWW

Od niedawna programowanie WWW jest jedną z szybciej rozwijających się i najbardziej obiecujących dziedzin internetowej informatyki. Programy działające via WWW podzielić można na dwa podstawowe rodzaje:

- Programy wykonywane na serwerze WWW
- Programy wykonywane po stronie klienta WWW

Przykładem języków wykorzystywanych w programach pierwszego typu są Java i JavaScript. Języki takie jak Java, dzięki temu, co stanowi ich istotę, czyli dzięki „niezależności od platformy” mają stać się językami przyszłości. Na razie jednak niewielka część dostępnych na rynku przeglądarek (w chwili obecnej **żadna** z najpopularniejszych) umożliwia obsługę apletów (aplikacji) zgodnych z oficjalną specyfikacją języka Java; co gorsza, mimo tego, iż oficjalnie firma Sun (twórca języka Java) ma prawo ustanawiać standard tego języka, istnieje kilka jego wersji różniących się w szczegółach. Nie polecam w związku z tym stosowania stron wykorzystujących Javę bez (przynajmniej) pozostawienia użytkownikowi możliwości wyboru strony bez tego „udogodnienia”. Ogólnie przy projektowaniu strony należy zdać się tylko i wyłącznie na język HTML, gdyż wszelkie (aktualne i przyszłe) języki wykonywane po stronie klienta mocno zależeć będą od używanej przeglądarki. Jeśli jednak zdecydujemy się na stosowanie któregoś z nich, dobrze jest stosować wersję obsługiwaną przez jak największą liczbę przeglądarek.

Inaczej wygląda sytuacja, jeśli chodzi o języki, które wykonują programy służące modyfikacji zawartości stron bądź do tworzenia dynamicznie dokumentów WWW. Dzięki zastosowaniu mechanizmu CGI (Common Gateway Interface), który zleca serwerowi wykonanie odpowiedniego programu (np. na podstawie umieszczenia w URL adresu skryptu wraz z parametrami) i przesłanie wyników jego działania do klienta (w przeciwieństwie do języków pierwszego rodzaju, które przesyłają klientowi *kod programu* do wykonania) są one niewidoczne dla przeglądarki, która otrzymuje dokładnie takie same dane, jak w wypadku przesyłania przez serwer strony będącej zwykłym dokumentem HTML. W wypadku, gdy serwer WWW pracuje pod kontrolą systemu operacyjnego U*X, do pisania

skryptów CGI można użyć dowolnego języka programowania, którego kompilator bądź interpreter zainstalowany jest na serwerze, takiego jak C, C++, tcl, Pascal, czy nawet języka skryptów shella; najczęściej jednak używany jest język Perl (w dowolnej wersji, najlepiej jednak w wersji 5). Skrypt CGI pobiera parametry umieszczone w adresie URL, który go uruchamia, i wyprowadza efekt swojego działania na standardowe wyjście, a serwer przesyła go do programu klienckiego. Najprostsze programy CGI w różnych językach wyglądają jak następuje:

Język C

```
#include <stdio.h>

main(){
printf("Content-type: text/plain\n");
printf("<HTML>");
printf("</HTML>");
}
```

Shell

```
echo "Content-type: text/plain"
echo
echo "<HTML></HTML>"
echo
```

Oba powyższe przykłady generują pusta stronę i wysyłają ją do klienta. Jak widać sama zasada działania CGI jest prosta. Wynikiem działania programu CGI powinno być to, co wysyłane jest przez serwer jako ostatnia część odpowiedzi. W systemach uniksowych wybór języka użytego do stworzenia skryptu nie ma znaczenia. Gorzej jest w wypadku komputerów pracujących pod kontrolą systemów MS Windows NT, w których nie istnieje pojęcie standardowego wyjścia. Do obsługi CGI w takich systemach stosowany jest interfejs WinCGI, pozwalający na dołączanie programów pisanych np. w VisualBasicu.

3. Standardy nieoficjalne

Najważniejsze spośród standardów nieoficjalnych starają się określić „dobre zwyczaje”, które warto stosować podczas projektowania strony. Część z nich omówiłem wcześniej, teraz pozostaje jedynie zwrócić uwagę na pozostałe. Warto przygotować stronę w taki sposób, aby jej ładowanie wraz z rysunkami zajęło niezbyt wiele czasu. Dokonywane na Zachodzie badania dotyczące preferencji użytkowników pokazały, iż w wypadku, gdy strona ładuje się w całości dłużej niż

20 sekund, większość potencjalnych jej odbiorców przerywa ładowanie, i rezygnuje z oglądania strony. Aby tak się nie działo, należy zadbać o to, by:

- pojedyncza strona nie była zbyt duża
- wielkość plików graficznych włączonych do strony pozwalała na ich szybkie ściągnięcie.

Lepiej jest stronę podzielić na niewielkie fragmenty, niż dopuścić, by jej wielkość nadmiernie wzrosła. Wielkość pojedynczego pliku graficznego powinna być najmniejsza z możliwych, nie większa jednak niż 40kb (optimum to 10-20kb).

4. Przykłady rozwiązań praktycznych

4.1. Jak rozpoznać przeglądarkę

Aby zaprojektowana przez nas strona mogła być czytelna zarówno na przeglądarkach przewidujących wspomaganie dla grafiki, jak i na tych, które takiego wsparcia nie mają, można stworzyć osobne wersje strony, przeznaczone dla obu rodzajów przeglądarek, można skorzystać jednak (co jest rozwiązaniem bardziej eleganckim), jeśli tylko istnieje możliwość dołączania skryptów CGI do stron, z tego, co oferuje protokół HTTP — rozpoznawania przy pomocy wysłanego przez przeglądarkę nagłówka User-Agent typu przeglądarki, i wysyłania odmiennych stron automatycznie, bez konieczności dokonywania wyboru przez użytkownika. Przykład konkretnego skryptu w języku Perl służącego do stworzenia dwóch wariantów strony:

```
use CGI;

$page = new CGI;

print $page->header;

if ($page->user_agent("lynx")){
print $page->start_html(-title=>'Strona dla przeglądarki tekstowej');
print << Koniec_txt;
Tutaj jakiś tekst,
hyperłącza, etc.
...

Koniec_txt
}
else{
print $page->start_html(-title=>'Strona dla przeglądarki graficznej');
print << Koniec_graf
<IMG SRC="ścieżka_do_pliku_rysunku">
dalsza część strony,
hyperłącza, etc.
```

```
...  
Koniec_graf  
};  
  
print $page->end_html;#13;
```

4.2. Jak pokazać polskie znaki

Aby poprawnie zobaczyć tekst zakodowany ISO-8859-2 w przeglądarce np. Netscape Navigator należy przełączyć za pomocą menu (View-Encoding-Central European (ISO-8859-2)) odpowiedni sposób pokazywania znaków. W wypadku, gdy oglądamy strony napisane w różnych językach, ciągle przełączanie kodowania może być niewygodne. Możemy, dopisując odpowiednie polecenie do nagłówka strony, poinformować przeglądarkę w jaki sposób kodowane są znaki w naszym dokumencie. oto ten fragment nagłówka:

```
<HEAD>  
  <META http-equiv="content-type" content="text/html;  
    charset=ISO-8859-2">  
</HEAD>
```

5. Podsumowanie

Standardy nie są bynajmniej stworzone po to, by komukolwiek przeszkadzać, wręcz przeciwnie, mają one za zadanie jedynie pomoc projektantom. Z tego powodu dziwna może być niechęć do ich stosowania. Być może powodem jest niechęć do ciągłej nauki i trzymania „ręki na pulsie”, czytania wielu umieszczonych w różnych zakamarkach Internetu dokumentów, sprawdzania efektów swej pracy na wiele różnych sposobów. Moim skromnym zdaniem katolickie serwisy w Internecie powinny być wzorem poprawnego i zgodnego ze standardami wykonania, co niestety nie zawsze ma miejsce. Trzeba jednak zdać sobie sprawę, iż, jeśli chodzi o standardy, właściwie nie ma wyboru. Problem „stosować czy nie stosować” nie istnieje. Chyba że ktoś woli amatorszczyznę od profesjonalizmu, czego absolutnie nikomu nie życzę.

Bibliografia

- [1] Materiały z seminariów: „Kościół w Polsce a Internet. Nowa Misja” i „Kościół w Polsce a Internet. Możliwości i perspektywy” Gliwice 1997
- [2] Materiały zamieszczone na Polskiej Stronie Ogonkowej:
<http://www.agh.edu.pl/ogonki>

-
- [3] RFC1945 Hypertext Transfer Protocol -HTTP/1.0.
 - [4] RFC2045 Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.
 - [5] RFC2046 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types.
 - [6] RFC2047 MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text.
 - [7] RFC2048 Multipurpose Internet Mail Extension (MIME) Part Four: Registration Procedures.
 - [8] RFC2049 Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples
 - [9] RFC2068 Hypertext Transfer Protocol -HTTP/1.1.
 - [10] RFC2070 Internationalization of the Hypertext Markup Language.
 - [11] Specyfikacje języka HTML: <http://www.w3.org>
 - [12] Stephen Spainhour & Valerie Quercia: Webmaster. Podręcznik administratora. Warszawa 1997