Chip Special. Poradnik Webmastera, nr 11(69) 2002, s. 69-71

XML a sprawa polska

Przemysław Kazienko
kazienko@pwr.wroc.pl

Z problemem polskich znaków w informatyce borykano się od bardzo wielu lat i nadal jest to kwestia do końca nierozwiązana. Niestety dotyczy to także języka XML i polskojęzycznych dokumentów w nim tworzonych.

1. Unikod — ujednolicony standard kodowania znaków

Jedną z cech języka XML jest jego „międzynarodowość". Wynika ona z tego, że zalecanym (w rekomendacji XML 1.0) standardem kodowania znaków jest międzynarodowy **Unikod** (*Unicode*). Z jego pomocą można zakodować znaki z praktycznie wszystkich, najważniejszych języków (pism) świata, w tym także np. hieroglify, cyrylicę, symbole wymowy, matematyczne, muzyczne czy ostatnio dodane znaki pisma filipińskiego. Aktualna wersja Unikodu 3.2 z marca 2002 roku, zawiera **95,221** (!) znaków. Liczba zakodowanych znaków w wersji 3.2 — w porównaniu z wersją 1.0 — wzrosła trzyipółkrotnie. Dla porządku, znaki zostały podzielone na 108 pism (grup znaków). Dostępna jest także pełna alfabetyczna lista znaków. Każdemu znakowi z każdego pisma odpowiada dokładnie jeden kod w Unikodzie. W ramach Unikodu istnieje kilka standardów zapisu kodów:

* UTF-8 — najpopularniejszy, w którym długość kodów jest zmienna i waha się od 1 do 4 bajtów czyli od 8 do 32 bitów
* UFT-7 — wersja 7-bitowa zgodna z RFC 2152; rzadko używana
* UTF-16 — format 16-bitowy, występujący w dwóch wersjach różniących się kolejnością bajtów
* UTF-32 — format 32-bitowy, w którym każdy znak zajmuje 32 bity.

Unikod jest standardem zalecanym dla języka XML, co znalazło swój wyraz w pierwszej specyfikacji tego języka — XML 1.0. Zalecono w nim wersję 2.0 Unikodu, gdyż taka wtedy była dostępna. Numer wersji nie ma jednak specjalnego znaczenia, ponieważ wszystkie ważne dla nas znaki zostały zawarte już w wersji 1.0.

Zalecanym rodzajem kodowania znaków w języku XML jest standard Unikod.

2. Polskie znaki w Unikodzie

Długość kodu w Unikodzie jest zmienna (w zależności od formatu kody mogą być z przodu uzupełniane) i zależy od tego jaką pozycję w świecie miał dany język. Znaki alfabetu angielskiego mieszczą się na jednym bajcie. Polskie znaki diakrytyczne (ąćęłńóśźż) są niestety mniej uprzywilejowane — zajmują dwa bajty i należą do grupy *Latin Extended-A*. Wyjątek stanowią 1-bajtowe „ó" oraz „Ó", należące do grupy *C1 Controls and Latin-1 Suplement* — patrz załączona tabela.

Tabela 1. Polskie znaki diakrytyczne w Unikodzie

Znak	Kod szesnastkowy	Kod dziesiętny	Nazwa angielska	Liczba bajtów UTF-8
ą	0105	261	Latin small letter A with ogonek	2
ć	0107	263	Latin small letter C with acute	2
ę	0119	281	Latin small letter E with ogonek	2
ł	0142	322	Latin small letter L with stroke	2
ń	0144	324	Latin small letter N with acute	2
ó	00F3	243	Latin small letter O with acute	1
ś	015B	347	Latin small letter S with acute	2
ź	017A	378	Latin small letter Z with acute	2
ż	017C	380	Latin small letter N with dot above	2
Ą	0104	260	Latin capital letter A with ogonek	2
Ć	0106	262	Latin capital letter C with acute	2
Ę	0118	280	Latin capital letter E with ogonek	2
Ł	0141	321	Latin capital letter L with stroke	2
Ń	0143	323	Latin capital letter N with acute	2
Ó	00D3	211	Latin capital letter O with acute	1
Ś	015A	346	Latin capital letter S with acute	2
Ź	0179	377	Latin capital letter Z with acute	2
Ż	017B	379	Latin capital letter N with dot above	2

Znaki Unikodu można umieszczać w dokumencie XML na trzy sposoby:

1. Podając wprost (za pomocą jakiegoś edytora lub konwertera) kod odpowiadający danemu znakowi.

2. Umieszczając jednostkę - encję (*entity*) z odpowiednim kodem dziesiętnym danego znaku, w postaci: `&#d;`, gdzie *d* to kod dziesiętny. Dla „ą” będzie to `&#261;`.

3. Za pomocą jednostki zawierającej kod szesnastkowy (heksadecymalnie), tj. w postaci: `&#xh;`, gdzie *h* — kod szesnastkowy. Dla „ą” mamy: `&#x105;`.

W obu ostatnich przypadkach należy koniecznie kod rozpocząć „&#” i zakończyć średnikiem.

Na marginesie można podać jedno z rozwiązań częstego problemu autorów dokumentów XML (a zwłaszcza XSLT), tj. sposób wstawiania spacji. W języku XML nie można zastosować znanej z języka HTML jednostki wbudowanej „` `”. jednym z rozwiązań jest wykorzystanie odpowiedniego znaku Unikodu, tj. zwykłej spacji (kod szesnastkowy 0020) lub tzw. „twardej spacji” (*no-break space*) — kod 00A0. Innymi słowy, wystarczy podać: „` `” lub „` `”.

Wróćmy jednak do polskich liter w standardzie Unikod. Aby je wprost umieścić w dokumencie XML (pierwszy, najlepszy ze sposobów), najprościej czynić to w dowolnym edytorze dla języka XML lub HTML, np. *XML Spy* czy *Pajączek*, który znaki z klawiatury zapamięta pod odpowiednimi kodami Unikodu. Większość edytorów HTML i XML wspiera standard Unikod, aczkolwiek najczęściej czyni to poprzez konwersję do i z „roboczego”

innego standardu. Unikod nie jest zwykle kodowaniem „roboczym" tych edytorów. Innym sposobem jest zastosowanie dowolnego konwertera, np. dostępnego online *Pierwszego Polskiego Internetowego Konwertera Kodowania Dokumentów Tekstowych* Piotra Trzcionkowskiego (http://www.trzcionk.priv.pl/programy/ppikkdt.html).

Dla przykładu, utwórzmy dokument XML zawierający polskie litery w Unikodzie podane na wszystkie wyżej wymienione sposoby. W standardowej przeglądarce plików dostępnej pod Windows, której zadamy wyświetlanie znaków ASCII z zestawu DOS, przykładowy dokument wygląda jak na pierwszym ekranie.

Rysunek 1 - Rys1.tif###: Zawartość dokumentu XML zawierającego polskie znaki, prezentowana przez przeglądarkę plików tekstowych systemu Windows.

Natomiast w dobrze skonfigurowanym edytorze *XML Spy* — w którym znaki podane wprost zostaną w obszarze roboczym przekonwertowane na Windows 1250 — dokument ten będzie widziany jak na drugim ekranie.

Rysunek 2 - Rys2.tif###: Dokument XML z polskimi znakami widziany w edytorze *XML Spy*.

W przeglądarce Internet Explorer 6.0, która (jak każdy parser) skomasuje podwójne spacje do jednej, usunie przejścia do nowej linii i odpowiednio zinterpretuje kody, przykładowy dokument ma postać jak na trzecim ekranie.

Rysunek 3 - Rys3.tif###: Dokument XML z polskimi literami w Unikodzie zinterpretowanymi przez Internet Explorera.

Jak widać podanie wprost kodów Unikod jest rozwiązaniem zdecydowanie najprostszym i daje dokument najbardziej czytelny i najkrótszy.

Standard Unikod jest obsługiwany zarówno przez Internet Explorer jak i Netscape Communicator. Oba od wersji 4.0. Wspierają go także wszystkie najważniejsze systemy operacyjne (Windows, Linux).

3. Inne standardy kodowania polskich znaków

Unikod to ogólnoświatowy, zunifikowany standard kodowania, co brzmi bardzo ładnie. Jednak nasza polska rzeczywistość jest bardziej skomplikowana. Wszystko komplikuje fakt, że istnieje kilka standardów kodowania polskich liter. Ma to niestety także wpływ na język XML.

Otóż, standard XML 1.0 wprawdzie zaleca Unikod, ale równocześnie dopuszcza także inne kodowania. Sposób kodowania określa się w deklaracji XML umieszczanej zawsze na początku dokumentu (nic nie może jej poprzedzić, nawet komentarz), w parametrze `encoding`, np.:

```
<?xml version='1.0' encoding='UTF-8' ?>
```

lub

```
<?xml version='1.0' ?>
```

Nie podanie parametru `encoding` **oznacza kodowanie Unikod UTF-8**. Obie powyższe deklaracje są więc równoważne.

Dopuszcza się zastosowanie także nieco dłuższej wersji Unikodu — UTF-16, poprzez:

```
<?xml version=”1.0” encoding=”UTF-16” ?>
```

Wszystko byłoby w porządku, gdyby nie możliwość zastosowania innego sposobu kodowania (tabela 2), np. ISO Latin-1, bez polskich znaków:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

lub ISO Latin 2 (ISO 8859-2), zawierającego polskie znaki:

```
<?xml version="1.0" encoding="ISO-8859-2"?>
```

Istnieje także kodowanie promowane przez firmę Microsoft, związane z systemem Windows: *Windows 1250*:

```
<?xml version="1.0" encoding="windows-1250"?>
```

To, czy dany rodzaj kodowania jest obsługiwany przez parser (program przetwarzający dokument XML) zależy od samego parsera i niektóre z nich mogą nie mieć zaimplementowanego np. standardu *Windows 1250*. Pewnym można być tylko tego, że wszystkie parsery powinny obsługiwać standard Unikod.

Tabela 2. Rodzaje kodowania polskich znaków

Rodzaj kodowania zawierającego polskie znaki diakrytyczne	Postać deklaracji XML

Unikod	`<?xml version='1.0' ?>`
	`<?xml version='1.0' encoding='UTF-8' ?>`
	`<?xml version='1.0' encoding='UTF-16' ?>`

ISO Latin 2	`<?xml version='1.0' encoding='ISO-8859-2'?>`
Windows 1250	`<?xml version='1.0' encoding='windows-1250' ?>`

Żeby było ciekawiej w praktyce możemy mieć do czynienia także z innymi rodzajami kodowania, np. z pochodzącym z systemu DOS — kodowaniem Mazovia i dla własnych potrzeb może ktoś dodać `encoding='mazovia'`. Taki standard nie będzie jednak obsługiwany przez inne, prócz własnych programy. Podobne przypadki możemy więc pominąć.

4. Polskie litery w nazwach elementów i atrybutów

Znaki narodowe można w zasadzie stosować w każdym miejscu dokumentu XML, aczkolwiek niektóre programy posiadają swoje ograniczenia w tym zakresie. Przykładowo, parser wbudowany w *XML Spy* (środowisko uznawane za jedno z najlepszych) nie toleruje niektórych polskich znaków w deklaracjach typów wyliczeniowych dla atrybutów umieszczanych w DTD.

Skoro znaki diakrytyczne można stosować wszędzie, to wszędzie, czyli także w nazwach elementów i atrybutów. Poprawny jest więc dokument:

```
<?xml version='1.0' encoding='UTF-8' ?>

<!-- Dokument XML z polskimi znakami w nazwach elementów i
atrybutów -->

<ŁÓDŹ żółw='tak' słoń='chyba nie'>
Ekskluzywna, superszybka łódź z żółwiem na pokładzie (w
charakterze maskotki), ale prawdopodobnie bez słonia
</ŁÓDŹ>
```

Element `ŁÓDŹ` jest wprawdzie poprawny, ale taka sytuacja niesie pewne zagrożenia. Zasadniczym problemem wynikającym z używania polskich liter jest to, że w dokumencie można wykorzystywać tylko jeden rodzaj kodowania znaków, wskazany w deklaracji XML. Nie można dla potrzeb tylko jednego ciągu znaków lub jakiegoś elementu zmienić kodowania.

Nie jest dopuszczalne stosowanie innego sposobu kodowania dla nazw elementów i atrybutów a innego dla ich treści i wartości w ramach tego samego dokumentu XML.

Wyobraźmy sobie, że treść elementu pochodzi z bazy danych albo z innych zewnętrznych źródeł. Jeżeli dokument będzie generowany automatycznie, to należy zadbać o to, by kodowanie nazw elementów (`ŁÓDŹ`) oraz atrybutów (`żółw`, `słoń`) było takie samo jak ich wartości. Może być z tym kłopot, zwłaszcza wtedy, gdy dopuścimy możliwość różnych sposobów kodowania. Jeżeli więc nazwy będą zakodowane w standardzie Unikod, zaś wartości w ISO Latin2, wtedy konieczne jest przekodowanie. Lepszym rozwiązaniem jest rezygnacja z polskich znaków w nazwach a w dokumencie stosowanie takiego kodowania, jakie wynika z danych (treści elementów i wartości atrybutów).

Jeszcze wyraźniej problem ten wystąpi w przypadku standardowych dokumentów XML, wykorzystywanych przez wiele różnych osób lub organizacji. Weźmy na przykład formularze podatkowe wysyłane do urzędów skarbowych przez podatników (firmy i osoby fizyczne). Ponieważ są one powszechne, więc powinno się w nich dopuszczać kilka sposobów kodowania. W przypadku zastosowania polskich znaków diakrytycznych w nazwach należałoby zatwierdzać osobne formaty standardowych formularzy dla każdego kodowania z osobna. Nie wydaje się to najlepszym rozwiązaniem. Jeżeli polskie znaki ograniczymy do treści elementów i atrybutów, wtedy dla właściwego przetwarzania wystarczy jedynie odpowiednia deklaracja XML na początku dokumentu.

Różne sposoby kodowania dodatkowo komplikują opracowywanie dokumentów powiązanych, np. transformacji XSLT, w których przecież występują zarówno nazwy elementów i atrybutów jak i dodatkowe teksty umieszczane w dokumencie wynikowym. Możemy zastosować inne kodowanie w transformacjach XSLT a inne w przetwarzanym dokumencie XML, gdyż są to odrębne pliki i osobnymi deklaracjami XML. Taka sytuacja wprowadza jednak dużo zamieszania przy modyfikacji obu dokumentów.

Z drugiej jednak strony polskie znaki zwiększają czytelność nazw. Istnieje więc dylemat: czy należy stosować polskie znaki w nazwach elementów i atrybutów, czy nie? Można sformułować następujące rozwiązanie tego dylematu:

* w systemach zamkniętych, zwłaszcza niezbyt dużych oraz w serwisach internetowych, w których nie ma powiązań pomiędzy dokumentami XML a bazami danych stosowanie polskich znaków w nazwach elementów i atrybutów może być przydatne;

* w systemach wymiany danych, z których korzysta wiele podmiotów oraz wtedy, gdy dokumenty XML pełnią rolę interfejsu do bazy danych, wydaje się, że lepiej zrezygnować z polskich znaków.

W przypadku, gdy nie musisz, to nie wykorzystuj polskich znaków diakrytycznych w nazwach elementów i atrybutów.

5. Internetowe źródła informacji

Anglojęzyczny serwis poświęcony standardowi Unikod (encyklopedia Unikodu): http://www.unicode.org

Polska strona poświęcona standardowi Unikod: http://www.unikod.pl

Polska Strona Ogonkowa: http://www.agh.edu.pl/ogonki/

Pierwszy Polski Internetowy Konwerter Kodowania Dokumentów Tekstowych (działający online) Piotra Trzcionkowskiego: http://www.trzcionk.priv.pl/programy/ppikkdt.html

Znaki z poszczególnych grup pism w Unikodzie: http://www.unicode.org/charts/

Znaki grupy Latin Extended-A zawierającej polskie litery (oprócz „ó” oraz „Ó”): http://www.unicode.org/charts/PDF/U0100.pdf

Alfabetyczna lista znaków Unikodu: http://www.unicode.org/charts/charindex.html

FAQ na temat UFT-8 oraz Unikodu: http://www.cl.cam.ac.uk/~mgk25/unicode.html

Statystyki znaków w Unikodzie: http://www.i18nguy.com/unicode/char-count.html

„Centrum XML” — serwis poświęcony językowi XML prowadzony na Politechnice Wrocławskiej: http://www.zsi.pwr.wroc.pl/xml

Specyfikacja języka XML 1.0 (wydanie drugie, poprawione redakcyjnie) zalecająca Unikod: http://www.w3.org/TR/2000/REC-xml-20001006