Wykorzystanie TEX4ht oraz XSLT do konwersji plików LATEXa

Włodzimierz Bzyl
Instytut Matematyki, Uniwersytet Gdański
80-952 Gdańsk, ul. Wita Stwosza 57
matwb@univ.gda.pl

Tomasz Przechlewski
Wydział Zarządzania, Uniwersytet Gdański
81-824 Sopot, ul. Armii Krajowej 119/121
tomasz@gnu.univ.gda.pl

Streszczenie

TEX4ht system is generally considered the best application for converting TEX files to HTML/XML format. TEX4ht system consists of three parts: style files which enhance existing macros with HTML, or DocBook, or TEI like features; tex4ht processor which extracts HTML (or DocBook/TEI) files from DVI files produced by tex; t4ht processor which is responsible for translating DVI code fragments which need to be converted to pictures; for this task the processor uses tools available on the current platform.

Out of the box, TEX4ht system is configured to translate roughly from plain, LATEX, ltugboat, ltugproc, Lecture Notes in Computer Science (llncs) formats to HTML/XML. Hovewer the conversion from a visual format to information oriented one cannot be done automatically and usually prior configuration of tex4ht is needed. Instead of configuring TEX4ht—which is not easy—we could use XSLT style-sheet to remap elements to reference XML format.

The paper introduces TEX4ht system. Selected problems of configuring the system and converting TEX/LATEXfiles to XML with TEX4ht are discussed.

Wprowadzenie

Powszechna dostępność różnorakich programów edytorskich spowodowała, że tworzenie dokumentów elektronicznych jest obecnie łatwe i tanie. Gwałtowny rozwój Internetu, jaki dokonał się w ciągu ostatnich lat spowodował z kolei, że bardzo tanie stało się także ich publikowanie, a to z kolei spowodowało, że udostępnianie informacji w tradycyjnej papierowej postaci nie jest już sposobem jedynym i dominującym. Publikowanie w Internecie wymaga udostępniania informacji w powszechnie akceptowanym formacie, którym w przypadku sieci WWW jest HTML/XML.

Jakie jest miejsce systemu TEX w nowej rzeczywistości? Według autorów ciągle jest on i będzie wygodnym systemem przygotowania dokumentów, w tym (a zwłaszcza) dokumentów naukowych i technicznych. Aby jednak TEX był w pełni „dopasowany do nowych czasów" konieczne jest m.in. rozwiązanie problemu konwersji dokumentów do formatów obowiązujących w sieci WWW. W chwili obecnej bowiem zamiana dokumentu TEX-owego do formatu HTML/XML nie jest zwykle sprawą prostą, tymczasem szybko ulepszana infrastruktura Internetu pozwala już teraz na publikowanie całkiem skomplikowanych typograficznie dokumentów, eliminując potrzebę stosowania „obcych ciał" w postaci formatów, takich jak PDF.

Niniejszy tekst stanowi wprowadzenie do problemu konwersji plików TEX do formatu XML oraz zawiera krótki przegląd publicznie dostępnych narzędzi wykorzystywanych w tym celu, ze szczególnym uwzględnieniem programu TEX4ht [4].

Konwersja plików źródłowych TEX-a

Do zamiany plików źródłowych TEX-a na format HTML/XML mogą być użyte następujące strategie:

1. Powtórne ręczne oznakowanie tekstu ewentualnie wspomagane częściową konwersją za pomocą takich narzędzi jak Perl czy Python.
2. Bezpośrednia konwersja plików źródłowych na format docelowy. W ten sposób działają takie programy, jak: `ltx2x` [5], `tex2html` [1], oraz `tralics` [3].

3. Przetwarzanie plików za pomocą TEX-a a następnie przetwarzanie wygenerowanych plików do docelowego formatu HTML bądź XML. W ten sposób działa system TEX4ht.
4. Generowanie plików XML bezpośrednio ze źródeł. Tak może działać system Omega.

Pierwsze z przedstawionych rozwiązań, acz prymitywne często może być rozwiązaniem najtańszym. Jeżeli zbiór dokumentów nie jest duży a ich oznakowanie niekonsekwentne i specyficzne, to większość gotowych rozwiązań z pewnością zawiedzie a koszt ich dostosowania może się okazać wyższy od zwykłego powtórnego oznakowania, np. po uprzednim usunięciu oryginalnej TEX-owej adiustacji.

Ponieważ zarówno pliki źródłowe TEX-a jak i wynikowy format konwersji są plikami tekstowymi najbardziej naturalne i oczywiste wydaje się drugie podejście do problemu. Niestety jego rozwiązanie nie sprowadza się tylko do przeczytania plik źródłowego, zidentyfikowania poleceń składu oraz ich wymiany na nazwy elementów czy atrybutów. Aby otrzymać prawidłowe wyniki, plik źródłowy powinien być zinterpretowany w sposób jaki to robi program `tex`.

Dlatego jedynym obiecującym sposobem konwersji jest wykorzystanie trzeciej strategii, ponieważ najtrudniejsze zadanie – interpretacja pliku źródłowego – jest wykonywana przez program `tex` [2].

Jak działa system TEX4ht?

System TEX4ht jest ciągle oprogramowaniem w fazie rozwoju, nie posiada dobrej dokumentacji a sposób jego konfigurowania nie jest prosty. Wszystko to utrudnia jego wykorzystanie przez użytkowników, zwłaszcza tych posługujących się innym językiem niż angielski.

Podstawowe części systemu TEX4ht to:

1. Program `tex4ht` konwertujący pliki DVI do XHTML.
2. Program `t4ht`, którego zadaniem jest wygenerowanie bitmap do tych fragmentów dokumentu, które będą reprezentowane za pomocą obrazków; sama konwersja do bitmap jest wykonywana za pomocą narzędzi zewnętrznych, takich jak ImageMagick czy PMB (polecenia, które należy wykonać aby wygenerować obrazki są umieszczone w pliku konfiguracyjnym `tex4ht.env`; w pliku tym są również zapisane ścieżki do hipertekstowych fontów wirtualnych, zob. pkt. 4 poniżej).
3. Plików z rozszerzeniem `.4ht` zawierających makra, które modyfikują działanie niektórych poleceń LATEX-a; zmodyfikowane makra umieszczają w pliku DVI dodatkowe informacje dla programu `tex4ht`, tak aby mógł on odtworzyć strukturę dokumentu źródłowego, konieczną do wygenerowania pliku w formacie XHTML.
4. Hipertekstowych fontów wirtualnych (ang. *hipertext text fonts*) zawierających informację o znakach umieszczanych w kodzie XHTML, np. w unikodowym foncie `plr.htf` w wierszu 129 znajdziemy informację, że litera 'Ą' ma być zamieniona na znak o kodzie szesnastkowym `#x0104` (unikod litery 'LATIN CAPITAL LETTER A WITH OGONEK').

Cały system możemy ściągnąć z witryny jego autora, Eitan'a Gurari [4]. Ponieważ w ostatnich dniach autor dodał do systemu wsparcie dla formatów plain, manmac i eplain jak również poprawił wiele błędów w plikach z makrami, więc zamiast instalować stare wersje 3) oraz 4) powinniśmy zainstalować nowe wersje dostępne z podstrony 'bug fixes' (archiwa `newt4ht.zip` i `htf.zip`). Sama instalacja jest bardzo prosta: pliki binarne umieszczamy w katalogu z plikami binarnymi systemu TEX, makra i fonty umieszczamy w jednym z drzewek TEX-owych. Ponieważ do wygenerowania ze źródła TEX-owego pliku HTMLpotrzeba trzech przebiegów `tex`-a oraz po jednym przebiegu programów `tex4ht` i `t4ht`, więc autor przygotował skrypty automatyzujące cały ten proces. Dla celów przykładu omówionego poniżej wystarczyło przystosować do języka polskiego skrypt `xhlatex` (w nowej wersji skrypt ten nazywa się `xplatex`)

Prosty przykład

System TEX4ht jest wysoce konfigurowalny, np. można go tak dopasować, aby program `tex4ht` generował plik w formacie RTF programu MS Word. Po instalacji system jest gotowy do generowania plików w formacie HTML i XHTML. Poniżej pokażemy jak wygenerować prosty plik w formacie XHTML w oparciu o plik źródłowy korzystający z klasy 'article'.

Nazwijmy plik przykładowy `00.tex`. Oto jego zawartość

```
\documentclass{article}
\usepackage{polski}

\title{TeX4ht}
\author{A. Nonim}
\date{Maj 2004}

\begin{document}
\maketitle
\section{Wstęp}
Artykuły opatrzone {\it takim} lub
```

```
{\it podobnym} tytułem, zaczną ukazywać
się {\bf po} konferencji w Bachotku.
\end{document}
```

Po kompilacji wykonanej za pomocą polecenia:

```
platex -translate-file=il2-pl 00.tex
```

i wydrukowaniu otrzymamy jedną stronę wyglądającą tak:

TeX4ht

A. Nonim

Maj 2004

1 Wstęp

Artykuły opatrzone *takim* lub *podobnym* tytułem, zaczną ukazywać się **po** konferencji w Bachotku.

Aby wygenerować plik XHTML ze źródła 00.tex wystarczy wykonać poniższe polecenie:

```
xplatex 00.tex mn.cfg
```

gdzie plik `mn.cfg` jest lokalnym plikiem konfiguracyjnym dla pliku `00.tex`. Jego zawartość jest typowa:

```
% Zmień domyślne rozszerzenie
% z '.html' na '.xhtml'
\Configure{html}{xhtml}
% Ustaw kodowanie w pliku wyjściowym na
% iso-8859-2
\Preamble{charset=iso-8859-2}
\begin{document}
\EndPreamble
```

Oto wynik kompilacji:

TeX4ht

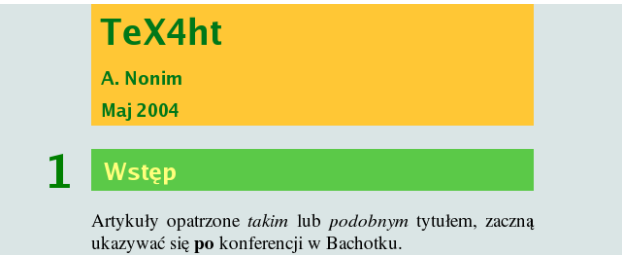
A. Nonim

Maj 2004

1 Wstęp

Artykuły opatrzone *takim* lub *podobnym* tytułem, zaczną ukazywać się **po** konferencji w Bachotku.

W sumie nic ciekawego – tylko font teraz to Times, a poprzednio był Polish Computer Modern. Spróbujmy teraz 'podrasować' nasz dokument umieszczając różne elementy w pudełkach o różnych kolorach.



Łatwo jest uzyskać taki efekt za pomocą kaskadowych arkuszy stylu. System TEX4ht użyje wcześniej przygotowanych arkuszy stylu jeśli w pliku konfiguracyjnym po wierszu z poleceniem `\Preamble` dopiszemy coś takiego:

```
\Configure{@HEAD}{}
\Configure{@HEAD}{\HCode{
<link rel="stylesheet" type="text/css"
  href="article.layout.bachotek2004.css"/>
<link rel="stylesheet" type="text/css"
  href="article.fonts.bachotek2004.css"/>
<link rel="stylesheet" type="text/css"
  href="article.colors.bachotek2004.css"/>
<link rel="stylesheet" type="text/css"
  href="\jobname.css" />\Hnewline}}
```

W terminologii TEX4ht nazywa się to konfigurowaniem 'Wrapper for the Document'. Opis tej i innych konfiguracji znajdziemy w pliku 'log' o ile polecenie kompilacji zmienimy w następujący sposób:

```
xplatex 00.tex "mn.cfg,info"
```

Zamiana na XML

Wprawdzie formalnie rzecz biorąc XHTML to też XML ale akurat ta aplikacja języka XML jest mocno zorientowana na prezentację. Wyraźnie to widać, jeżeli zajrzymy do środka pliku `.xhtml` wygenerowanego przez program TEX4ht:

```
<h2 class="titleHead">TeX4ht</h2>
<div class="author" ><span
class="plr-12">A. Nonim</span></div>
<div class="date" ><span
class="plr-12">Maj 2004</span></div>
</div>
<h3 class="sectionHead"><span
class="titlemark">1 </span> <a
 id="x1-10001"></a>Wstęp</h3>
<!--l. 15--><p class="noindent">Artykuły
opatrzone <span class="plti-10">takim
</span>lub <span class="plti-10">podobnym
</span>tytułem, zaczną ukazywać się <span
class="plbx-10">po </span>konferencji
w Bachotku.
</p>
```

TEX4ht Pozwala na zamianę TEX-owego pliku źródłowego na inne, bardziej zorientowane na oznakowanie semantyki/struktury, aplikacje standardu XML, takie jak: DocBook czy TEI. Służą do tego skrypty `dblatex` i `teilatex` (oraz ich warianty).

Przykładowo aby wygenerować z pliku TEX dokument XML zgodny z deklaracją DocBook należy wykonać następujące polecenie:

```
dblatex plik.tex
```

Z dokładnością do sposobu kodowania znaków otrzymamy w wyniku dokument XML formalnie zgodny z deklaracją DocBook.

Aby dokument był poprawny należy w wierszu poleceń dodać odpowiedni plik konfiguracyjny. Jako minimum powinien on określić kodowanie pliku wynikowego:

```
\Configure{html}{xml}
\Preamble{}
\Configure{VERSION}
   {\HCode{<?xml version="1.0"
     encoding="iso-8859-2"?>\Hnewline}}
\begin{document}
\EndPreamble
```

Domyślnie TEX4ht oznakowuje wiele fragmentów dokumentu wprawdzie poprawnie formalnie, ale w sposób nietypowy. Przykładem może być tytulatura, w której do oznakowania tytułu, autora i daty jest wykorzystywany element `<note>` z atrybutem `role` o odpowiedniej wartości. Konwersja dokumentu `00.tex` daje w rezultacie następujący fragment:

```
<para role="maketitle">
<note role="title"><para>TeX4ht</para></note>
<note role="author"><para><personname>
 <othername>A. Nonim</othername>
  </personname></para></note>
<note role="date"><para>Maj 2004</para>
</note> </para>
```

Zamiast, np:

```
<articleinfo>
 <title>TeX4ht</title>
 <author><personname><othername>
   A. Nonim</othername></personname></author>
 <date>Maj 2004</date> </articleinfo>
```

Na podobne uproszczenia można się natkąć także w innych miejscach (konwersja środowiska `figure`).

Wygodnym sposobem poprawienia struktury otrzymanego pliku `.xml` jest wykorzystanie do tego szablonów XSLT[1]. Przykładowo poniżej przedstawione dwie reguły szablonu:

```
<!-- zawartość elementu para z atryb.
  role='maketitle sformatuj jak trzeba: -->
 <xsl:template match="para[@role='maketitle']">
   <articleinfo><author>
    <xsl:value-of select="note[@role='author']"/>
      </author>
         ... itd...
   </articleinfo>
    </xsl:template>
<!--
 każdy inny element skopiuj bez zmiany: -->
    <xsl:template match="@* | node()">
      <xsl:copy>
        <xsl:apply-templates select='@*'/>
        <xsl:apply-templates />
      </xsl:copy>
    </xsl:template>
```

przepisują cały dokument XML oprócz elementu `<para>` z atrybutem `role` o wartości `maketitle`, którego formatowanie określa pierwsza z ww. reguł. Dodanie do powyższego programu reguły:

```
<xsl:template match="comment()" />
```

Powoduje, że dodatkowym efektem przetworzenia dokumentu będzie usunięcie komentarzy, których całą masę TEX4ht wstawia automatycznie. Jeżeli jeszcze do elementu `<output>` dodamy atrybut `indent` o wartości `yes`, to struktura dokumentu będzie elegancko zaznaczona za pomocą odpowiednich wcięć akapitowych (czego TEX4ht nie potrafi).

Zakończenie

System TEX4ht w połączeniu z językiem XSLT jest sprawnym narzędziem pozwalającym na praktyczną realizację zadania konwersji plików `.tex` do formatu XML.

Literatura

[1] Piotr Bolek. LaTeX2html. *Biuletyn GUST*, 10:37–48, 1998. `http://www.gust.org.pl/PDF/BIUL/10/l2html.pdf`.

[2] Michel Goossens and Sebastian Rahtz. *LATEX Web Companion*. Addison-Wesley, 2001.

[3] Jose Grim. Tralics. In *EuroTEX Proceedings*, pages 38–49, 2003. `http://www-sop.inria.fr/miaou/tralics`.

[4] Eitan Gurari. Tex4ht: LATEX and TEX for hypertext. `http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn.html`, 2004.

[5] Peter R. Wilson. LTX2X: A LATEX to X Auto-tagger. `http://www.ctan.org/tex-archive/support/ltx2x`, 1999.

[1] Oczywiście zamiast XSLT można konfigurować TEX4ht poprzez dodawanie odpowiednich wpisów do pliku konfiguracyjnego.