Bijlage 14

Conversie van anyTEX naar HTML met TEX4ht

Erik Frambach
Rijksuniversiteit Groningen
email: E.H.M.Frambach@eco.rug.nl

abstract

Gurari's TEX4ht-programmatuur maakt het mogelijk om in TEX geschreven teksten (plain TEX, LATEX, of wat dan ook) vrij eenvoudig te converteren naar HTML. Als voorbeeld nemen we de 4TEX-handleiding die in LATEX is geschreven.

keywords

conversie, html, www, TEX4ht

Introductie

Vrijwel iedereen zal inmiddels wel een idee hebben wat HTML is. Heel kort samengevat is HTML een standaard om teksten voor het World Wide Web in op te maken. Qua structuur lijkt het erg veel op (La)TEX zodat conversie niet al te moeilijk is.

In TEX schrijf je bv. '`{\it cursief}`', in HTML schrijft je '`<it>cursief</it>`'.

In LATEX schrijf je bv. '`\chapter{hoofdstuk}`', in HTML wordt dat '`<h1>hoofdstuk</h1>`'.[1] Ook in HTML kun je tekst over een willekeurig aantal regels verdelen net als in TEX, echter een nieuwe alinea moet expliciet aangegeven worden met '`<p>`'. Een lege regel heeft geen speciale betekenis zoals in TEX.

Uiteraard zijn er nog veel meer subtiele en minder subtiele verschillen maar die komen we vanzelf tegen in de rest van dit verhaal.

Principes

TEX4ht kan samenwerken met willekeurige TEX formats doordat hij zelf helemaal geen TEX-code interpreteert zoals bv. LATEX2html, maar vrijwel alles overlaat aan TEX zelf. De truuk hierbij is dat `tex4ht.sty` in de dvi-file '`\special`'s toevoegt die later door TEX4ht als besturing worden gebruikt voor de conversie naar HTML.

Vereenvoudigd komt het erop neer dat bv. aan het `\section`-commando '`\special{<h2>}`' en '`\special{</h2>}`' wordt toegevoegd. TEX4ht kan die later gemakkelijk uit de dvi-file vissen.

Uiteraard is het in werkelijkheid veel ingewikkelder omdat o.a. de inhoudsopgave ook goed moet komen. En natuurlijk wil je vanuit de inhoudsopgave meteen naar een sectie kunnen springen. Er komt dus heel wat boekhouding bij kijken.

Het grote voordeel van deze aanpak is dat nummering, referenties en (niet in de laatste plaats!) macro's door TEX al zijn verwerkt. TEX4ht kijkt uitsluitend naar wat TEX in de dvi-file stopt.

Noodzakelijke aanpasssingen

Om met TEX4ht te werken moet er om te beginnen een stukje code toegevoegd worden aan de tekst. In het geval van een LATEX-tekst gaat dat als volgt:

```
\documentclass{rapport1}

\input tex4ht.sty
\Preamble{htm,3,fonts}
\begin{document}
\EndPreamble

\end{document}
```

De style file `tex4ht.sty` wordt dus niet via `\usepackage` geladen. De reden daarvoor is dat de style file niet strikt LATEX is maar in principe met elk format kan samenwerken.

Het `\Preamble`-statement bepaalt dat de file extensie 'htm' wordt (ik had ook 'html' kunnen specificeren, maar MS-DOS-gebruikers hebben daar moeite mee). De '3' geeft aan dat ik niet één lange lap HTML wil, maar een splitsing tot op drie niveau's: 'chapter', 'section' en 'subsection'. Daardoor krijg ik dus een groter aantal HTML-bestanden die stuk voor stuk veel kleiner zijn en dus meer geschikt voor online publicatie. TEX4ht zorgt daarbij zelf voor de noodzakelijke structuur met verwijzingen naar de samenstellende delen.

De optie 'fonts' geeft aan dat ik wil dat TEX4ht zo veel mogelijk de gebruikte fonts (d.w.z. italic, bold, sansserif) in de HTML-versie respecteert.

1. Omgekeerd is conversie van HTML naar LATEX ook eenvoudig. HTML kan ook direct door TEX geïnterpreteerd worden, bv. met Carlisles LATEX-package 'typehtml'.

Met deze eenvoudige aanpassing kunnen we al meteen aan de slag, doordat `tex4ht.sty` al voorgeprogrammeerd is om met LATEX samen te werken.

TEX4ht draaien

Als we de TEX-file met de bovenbeschreven aanpassing compileren krijgen we een dvi-file die niet meer geschikt is om 'gewoon' te bekijken of afdrukken. Het is daarom verstandig om te beginnen met alles wat bij de tekst hoort te kopiëren naar een nieuwe directory en vandaaruit met TEX4ht te werken.

De nieuwe dvi-file kunnen we nu door TEX4ht halen. Die genereert nu in één keer de HTML-code, maar ook nog een paar extra bestanden. De belangrijkste daarvan is de 'ivd' file (het omgekeerde van 'dvi'). In dat bestand staan die stukken uit de dvi-file die niet door TEX4ht c.q. niet in puur HTML kunnen worden weergegeven. Dat zijn met name figuren, maar ook die geaccentueerde letters die in HTML niet gedefinieerd zijn. Die staan allemaal op afzonderlijke pagina's in de ivd-file.

Voor de interne administratie genereert TEX4ht ook bestanden met de extensie 'xre' (voor kruisverwijzingen) en 'otc' (voor de inhoudsopgave). Daar hoef je zelf verder niet naar om te kijken.

[next] [prev] [prev-tail] [tail] [up]

Chapter 3 Installing 4TeX

3.1 Introduction
3.2 Typography conventions
3.3 TeX and 4TeX
3.4 A Historical Note, or `What's in a Name?'
3.5 Principles of 4TeX installation from CD-rom
3.5.1 Installation from CD-rom
3.5.2 Installation on your hard disk
3.5.3 Fine-tuning 4TeX to your taste
3.5.4 Network setup
3.5.5 Directory set up
3.6 Starting 4TeX

[next] [prev] [prev-tail] [front] [up]

Een partiële inhoudsopgave

Die ivd-file kan vervolgens naar PostScript worden omgezet, bv. met DVIPS. Dat PostScript-bestand wordt op zijn beurt door Ghostscript verwerkt tot afzonderlijke PCX-plaatjes per pagina.

Nou ja plaatjes, het zijn echt hele pagina's, zij het op lage resolutie, bv. 110 dpi, want de uitvoer is bedoeld voor presentatie op computer-beeldschermen.

Die PCX-plaatjes moeten eerst van hun ongewenste witruimte worden ontdaan ('cropping'). Daarvoor kan het programma 'display' gebruikt worden. Dat programma kan meteen de conversie doen van PCX naar GIF, een graphics-formaat waar elke WWW-browser mee overweg kan.[2]

Dan volgt er nog een laatste slag waarbij de GIF-plaatjes 'transparant' worden gemaakt. Daardoor krijgen de plaatjes bij weergave door een WWW-browser toch de achtergrondkleur van de HTML-pagina. Het programma 'giftrans' zorgt daarvoor.

Uiteraard heeft TEX4ht zijn boekhouding netjes bijgehouden zodat hij weet welk GIF-plaatje op welke plek in een HTML-bestand moet worden gebruikt.

Overigens moet gezegd worden dat deze procedure niet echt eenvoudig is maar wel goed te automatiseren. In 4TEX versie 4 is de hele procedure met de druk op een knop in werking te stellen. De HTML-versie van de 4TEX-handleiding op de cdrom is direct door 4TEX geproduceerd zonder 'handwerk'.

Ook moet gezegd worden dat de procedure tijdrovend is. TEX4ht is zelf erg lang bezig met het verwerken van een dvi-file, maar ook het genereren van de plaatjes kan veel tijd kosten. Echter een HTML-versie genereren van een compleet boek doe je niet elke dag.

Verfijningen

Ofschoon de eerste resultaten lang niet slecht zijn kan er nog heel wat verbeterd worden.

In het geval van de 4TEX-handleiding heb ik daarom de volgende aanpassingen gedaan.

Na de `\Preamble` heb ik toegevoegd:

```
\Configure{centerline}%
  {\HCode{<CENTER>}}{\HCode{</CENTER>}}

\let\Xincludegraphics=%
  \includegraphics
\def\includegraphics[#1]#2{%
  \hbox{\Picture+[pict]{}%
  \Xincludegraphics[#1]{#2}%
  \EndPicture}}
```

Daarmee zorg ik ervoor dat `\centerline` ook in HTML wordt ondersteund. Verder breid ik `\includegraphics` uit zodat TEX4ht weet wat daarmee moet gebeuren: een plaatje ervan maken. Merk op dat ik met deze simpele definitie

2. Wanneer je het programma 'display' onder 4TEX onder Windows NT draait 'lekt' het geheugen. Daardoor had ik voor de conversie (56 plaatjes achter elkaar 'croppen') zo'n 150 MB geheugen nodig! Na verlaten van 4TEX is alles weer normaal. Wees gewaarschuwd.

mezelf verplicht tot het geven van opties in vierkante haken. Het kan mooier maar dat is slechts syntactische suiker.

Een volgende verfijning is het verwijderen van `\tableofcontents`. TEX4ht genereert die zelf al, en twee TOC's is te veel van het goede. Ook al begint het boek met een voorwoord, de inhoudsopgave komt altijd helemaal bovenaan. Het voorwoord komt daarom nu *wel* in de inhoudsopgave, wat ook logisch is.

Ook bibliografieën moeten anders verwerkt worden. In plaats van

```
\bibliographystyle{plain}
\bibliography{mybib}
```

moet je nu opnemen:

```
\input myfile.bbl
```

Uiteraard moet de bbl-file uit een eerdere gewone TEX- en bibTEX-run beschikbaar zijn.

Bibliography

[1] P.W. Abrahams, K.A. Hargreaves, and K. Berry. *TeX for the impatient*. Addison-Wesley, 1990.

[2] Adobe Systems Incorporated. *PostScript Language Tutorial and Cookbook*. Addison-Wesley, 1985.

[3] Adobe Systems Incorporated. *PostScript Language Reference Manual*. Addison-Wesley, 1990.

[4] S. von Bechtolsheim. *TeX in Practice: 1. Basics*. Springer-Verlag, 1993.

Een stukje bibliografie

Strikt genomen kan in HTML de verlaagde E uit het TEX-logo wel weergegeven worden, maar omdat exacte positionering (kerning) niet mogelijk is wordt het altijd lelijk. Dan maar liever gewoon 'TeX'. Voor het LATEX-logo geldt dat nog sterker. Dat is gelukkig ook precies wat TEX4ht doet. Voor de liefhebbers: het LATEX-logo zou in HTML ongeveer zo uit moeten zien (op 1 regel zonder spaties!):

```
L<SUP><FONT SIZE=-2>A</FONT></SUP>
T<SUB><FONT SIZE=+0>E</FONT></SUB>X
```

maar het hangt maar net van je browser en van het lettertype af of dit ergens op lijkt of niet.

Het LATEX-logo in html: mooi of niet?

Omdat we nu een interactief document maken is het aardig om email-adressen en verwijzingen naar WWW-pagina's meteen als 'link' weer te geven:

```
\def\email#1{%
  \HCode{<A HREF="mailto:#1">}%
  \texttt{#1}\HCode{</A>}}
\def\url#1{%
  \HCode{<A HREF="#1">}%
  \texttt{#1}\HCode{</A>}}
```

Daar is natuurlijk wat meer kennis van HTML voor nodig, maar het geeft aan hoe eenvoudig je zelf m.b.v. `\HCode` zelf HTML-code kan toevoegen. Een ander aardig voorbeeld is een andere definitie voor de toets-icoontjes die in de papieren handleiding worden gebruikt (bv. Esc). Van die icoontjes zou ik plaatjes kunnen maken, maar dat bleek ondoenlijk. TEX4ht beschouwt namelijk elke instantie van een plaatje als uniek en zou dus honderden plaatjes gegenereren terwijl er maar zo'n 30 unieke icoontjes zijn (dit geldt overigens ook voor bv. geaccentueerde letters). De volgende oplossing is daarom beter:

```
\def\toetsfont{\Large\sf}
\def\toets#1{%
  \HCode{<FONT COLOR="\#FF0000">}%
  {\toetsfont[#1]}%
  \HCode{</FONT>}}
```

In plaats van een zwart omkaderd teken krijg je nu de toets tussen vierkante haken, helemaal in het rood. Op het beeldscherm is dat heel effectief en natuurlijk veel sneller dan met plaatjes.

Een andere eigenaardigheid die verbeterd moest worden was de interpretatie van LATEX's '`\\`' commando. Dat bleek TEX4ht helemaal te negeren. Geen nood. Het HTML 'break'-commando kan zo toegevoegd worden:

```
\let\omslag=\\
\def\\{\omslag\HCode{<BR>}}
```

Merk op dat het commando's '`\\*`' nu niet meer werkt, maar die optie is voor de HTML-versie toch overbodig. Voor '`\\[xx]`' geldt dat in mindere mate omdat die wel eens misbruikt wordt om witruimte tussen 'alinea's' te maken. Die heb ik daarom vervangen door `\par`.

Een ander aspect dat juist opvallend gemakkelijk door TEX4ht wordt opgelost is tabellen.

Door in alle gevallen domweg standaard 'tabular'-omgevingen te gebruiken gaat alles vanzelf goed. TEX4ht genereert een HTML `<TABLE>` omgeving zodat die snel en betrouwbaar door een WWW-browser getoond kan worden. LATEX2html waagt zich daar niet aan en genereert een plaatje.

Verder kwam de conversie vooral neer op het verwijderen (voor zover nodig) van overbodige statements: `\clear(double)page`, `\parskip`, `\parindent`, `\textheight` en andere opmaakcommando's hebben geen zin meer.

	n^*	R^*	$EK^{[0,L]}(\pi^{R^*}(n^*))$
case 1	0	∞	0.8
case 2	1	11.4	1.7
case 3	2	0.6	5.0
case 4	4	1.2	6.4
case 5	6	1.7	9.8

Een willekeurige tabel

Met formules gaat TEX4ht heel zorgvuldig om. Voor zover mogelijk gebruikt hij HTML, en waar nodig laat hij TEX/DVIPS plaatjes genereren.

Wanneer we bv. $\sin x^2 \sum_1^\infty \delta x$ als volgt invoeren:

```
$\sin x^2 \sum_1^\infty \delta x$
```

dan genereert TEX4ht daarvoor de volgende HTML-code:

```
sin x<SUP >2</SUP>
<FONT FACE="SYMBOL">a</FONT>
<SUB >1</SUB>
<SUP ><FONT FACE="SYMBOL">Y</FONT>
</SUP><FONT FACE="SYMBOL">d</FONT>x
```

Keurig. Zetten we dezelfde formule in een equation-omgeving dan genereert TEX4ht een plaatje, waarin natuurlijk ook het formulenummer terugkomt.

Conclusie

De resultaten die met TEX4ht te bereiken zijn mogen gezien worden. Toch zijn er nog een paar punten die verbeterd zouden kunnen worden.

- Je hebt geen controle over de plaats (en vormgeving) van de inhoudsopgave.
- TEX4ht genereert zelf [up], [front], [tail], [prev-tail], [next] en [prev] links, zowel boven- als onderaan elke pagina. Onderaan de pagina komen ze echter soms direct achter de lopende tekst, dus zonder witruimte.
- De HTML-code is niet erg fraai. WWW-browsers hebben daar geen enkele moeite mee, maar als je zelf toch iets wilt aanpassen is het hinderlijk.
- Plaatjes worden niet hergebruikt ook al zijn ze volstrekt identiek.
- De fontdefinitie-bestanden die TEX4ht meelevert zijn goed bruikbaar maar geven soms toch onvoldoende informatie om de conversie goed te laten verlopen. Het beste kun je je helemaal beperken tot de Computer Modern familie of de 'mathptm' fonts.
- De uitvoer van MakeIndex wordt (nog) niet ondersteund.

Maar laten we niet vergeten dat dit programma nog maar heel kort bestaat. Bovendien heeft het ten opzichte van zijn meest direct concurrent LATEX2html ook duidelijke voordelen:

- Het systeem is LATEX-onafhankelijk: het kan samenwerken met elk TEX-systeem.
- Het systeem is niet Unix-afhankelijk. Dat is LATEX2html inmiddels ook niet meer zo erg, maar toch.
- Veel lastige zaken (macro's!) worden aan TEX overgelaten. Wie weet het nou beter dan TEX?
- Nummering van hoofdstukken, secties, formules, tabellen, figuren etc. blijft intact.
- Tabellen worden keurig in HTML vertaald.
- Input-encoding doet er niet toe, want TEX regelt dat.

De resultaten van de conversie van de 4TEX-handleiding staan op de 4allTEX cdroms (zie `\4texdoc\4texdoc.htm`) en (enigszins aangepast) op de NTG WWW-server: `http://www.ntg.nl/4allcd/4texdoc/4texdoc.htm`

Uiteraard staat het hele TEX4ht-pakket ook op de 4allTEX cdroms (zie disk2, `distrib\tex4ht`) en is het ingebouwd in 4TEX.

Via Gurari's WWW-pagina `http://www.cis.ohio-state.edu/~gurari/` is altijd de nieuwste versie van TEX4ht op te halen. Daar zijn ook verschillende mooie voorbeelden te zien van conversies. Kijk vooral eens naar `http://www.cis.ohio-state.edu/~gurari/TeX4ht/mn162.html` waarin wordt weergegeven hoe bekende public domain TEX- en LATEX-documenten naar HTML geconverteerd kunnen worden. Zeer leerzaam.