From Text to Hypertext:
A Post-Hoc Rationalisation of LaTeX2HTML

Nikos Drakos[*],
Computer Based Learning Unit,
University of Leeds,
Leeds LS2 9JT, UK.

phone: +44 (0)532 - 334626
fax: +44 (0)532 - 334635
email: `nikos@cbl.leeds.ac.uk`

A hypertext version of this document is also available[†].

April 18, 1994

Abstract

The conventional hypertext authoring framework compels authors to represent their material as an interconnected network of nodes and links. Apart from the difficulties that this alone entails, the situation with HTML is even more problematic since the author is also responsible for mapping the abstract network model onto the computer file system. This is likely to hinder the widespread adoption of HTML by information owners who are finding it difficult not only to create but also to maintain coherent documents with complex interconnection topologies.

In this paper it is argued that familiar document forms such as books, manuals, articles, reports etc. often contain sufficient structural and cross-referential cues with which to build a rich hypertextual structure. It is shown how this structure can be automatically extracted and then realised as a collection of HTML files which can be explored using generated navigation panels. The conversion process and the advantages of this approach are illustrated with interactive examples using the LaTeX2`HTML` converter. Other unique features of LaTeX2`HTML` - mathematical equations and "conditional text" - are also discussed.

Allowing authors to work with familiar metaphors and tools without compromising the flexibility afforded to them by the target hypertext system and delivery mechanism is perhaps the main reason for the growing popularity of text to hypertext conversion tools.

---

[*] http://cbl.leeds.ac.uk/nikos/personal.html

[†] http://cbl.leeds.ac.uk/nikos/doc/www94/www94.html

1 Introduction

The growth in the popularity of the WorldWide Web (WWW) during the past year has been phenomenal [9, 8]. One of the reasons for its success is its adherence to the principle of "Universal Readership" which simply states that once information is available, it should be accessible from any type of computer, in any country, using only one simple program [3].

This principle has been realised in the elegant way in which WWW blends hypermedia techniques with networked information retrieval to create a dynamic "docuverse" which can be traversed by selecting textual or iconic active areas on a screen, or searched via query mechanisms. Behind the scenes there are a number of protocols at work to facilitate the exchange of information between "client" and "server" programs including a universal addressing scheme (URL), the HyperText Transfer Protocol (HTTP), and the HyperText Markup Language (HTML) [11].

Today different communities are beginning to to benefit from the principle of Universal Readership by being able to tap into a vast pool of information sources. But there is also a demand for tools that support a complimentary principle which one might call "Universal Authorship", that is tools that make it easier for information owners to make their material available via the WWW as HTML documents[1].

In this paper some of the problems associated with HTML authoring are examined. In particular, the difficulty in divorcing the conceptual model of a hypertext document from low level delivery details is highlighted. It is then suggested that text to hypertext conversion tools may help to overcome many of the HTML authoring problems. This suggestion is substantiated with counter-arguments and solutions for a number of common objections against the use of text to hypertext conversion. The final section consists of some interactive examples illustrating these solutions in the context of the LaTeX2`HTML` conversion tool.

2 HTML Authoring

Making information available on the WorldWide Web in a way that exploits its capabilities usually involves the creation of interconnected networks of "nodes" or "pages" . Each node usually represents a concept or an idea and can contain structured text, images and local or remote hypertext links. These may point to other nodes, to different types resource types (eg audio or video), to other Internet services (eg ftp, wais, usenet, etc.) and may activate arbitrary external programs.

The first hurdle that budding WWW authors must overcome is that each node or page must be written in HTML. However, this is not so bad as there are macro extensions available for most popular word-processing systems (eg for WordPerfect, MSWord etc) some of which are menu driven (eg for Emacs) or even WYSIWYG (TkWWW) [11]. These HTML aware word-processors and editors can help someone overcome the problem of learning or remembering the HTML syntax.

A more difficult problem in HTML authoring is not the creation of individual nodes but the building of "coherent" network topologies. In common with other hypertext systems it is an author's responsibility to organise their material in a sensible way and to assist potential readers to explore it with appropriate navigation aids. Such aids help readers understand where they are in the network,

---

[1] Although documents in any format can be delivered using the WWW, HTML is necessary for taking full advantage of hypermedia capabilities.

where they came from, how to get to another place, and in general help in reducing the problems of disorientation and cognitive overload [1].

What complicates things further in the case of HTML is that in most cases each document network has to be created and manipulated as a set of interconnected *physical files*. An evolving network realised as physical files is a very inflexible structure to deal with, which leads to a number of problems:

- It is difficult to visualize, so unless an overall design is already mature or readily available in another medium the author herself may begin to suffer from disorientation.
- Structural changes especially after navigation aids are in place carry a lot of overhead as the author tries to maintain the integrity of existing hypertext links and the coherence of the network in general.

To gain an appreciation into the complexity of the task consider a document with a small number of nodes, an intuitive hierarchical structure and rudimentary navigation facilities such as the HTML version of this paper. Figure 1 shows a possible "node map" with the number of the outgoing links from each node. An average of 16 hypertext links out of a total of over 350 would have to be redirected or created *in several separate files* after a node is deleted, added or moved.

Figure 1: A map of the HTML nodes generated from this document and the number of outgoing hypertext links in each node.

In addition there are a number of other issues which are frustrating the efforts of many information owners trying to use HTML. There is often a need to create and deliver documents in different formats using different media, eg high quality paper-based versions of an electronic document that conforms to precise typesetting rules. Currently, alternative versions of the same document have to be created and maintained manually - a laborious and error-prone process. This situation may improve with the arrival of "HTML style-sheets" but no timescales are yet available. A similar problem occurs in the case of already existing documents in other formats which must be converted, often manually to HTML.

Another problem arises from the fact that HTML provides facilities for structuring the contents of each node, leaving the presentation details to each browser and out of the reach of the author. This makes HTML portable and easier to learn but frustrates many authors who would like more control over the presentation of their material. This is a common source of complaints especially where the presentation information is as important as the actual text (eg with mathematical equations).

3 Text To HyperText Conversion

A successful system which can convert documents written using a common word-processing system for paper-based delivery into HTML “webs” can eliminate many problems: the need to learn a new hypertext language simply disappears as documents can be written in a familiar word-processing format; the material is organised coherently around time honored metaphors which are shared both by the author and the reader; the author is insulated from the intricacies of dealing with physical files; both an electronic version and a high quality paper version of a document is available from a single source; and existing documents become immediately available.

3.1 Common Objections to Automatic Conversion

Given their many attractive benefits, a reasonable question to ask is “why aren't there many text to hypertext conversion systems?” This is principally because they are perceived as *difficult to built* and *limited in their scope*. In what follows we try to dispel these perceptions.

3.1.1 Word-processing formats contain presentation information that cannot be reproduced in HTML

To deal with this problem a conversion system must carefully distinguish between presentation information that can be safely ignored without loss of meaning (eg page layout) and presentation that can be reproduced in HTML (eg bulleted list items).

A more subtle problem however occurs in the case of meaningful layouts which cannot be reproduced eg special symbols, mathematical equations, colored text, included images, etc. A novel solution first introduced in the LaTeX2`HTML` conversion system relies on the ability of some HTML browsers to display images embedded in the main text. Each part of a source document containing meaningful presentation information is extracted from the main text, and then placed back in the generated HTML document *after being converted into an image*. During this process care is taken to preserve contextual information that may affect the contents of each image (eg automatic figure or equation numbering, citations etc). See the example on mathematical equations (Section 4) for an illustration of this technique.

For additional flexibility the LaTeX2`HTML` system allows users to modify the default settings on whether some presentation information which has no HTML equivalent should be ignored or whether it should be converted into an embedded image.

3.1.2 Conventional paper-based documents have a linear structure and are therefore unsuitable for conversion to the hypertext form

Many conventional printed document forms such as manuals, articles, reports, dictionaries etc. often contain many “structural” and “cross-referential” cues that allow them to be used in a non-linear fashion [1]. The basic structure is often hierarchical and consists of combinations of parts, chapters, sections, subsections etc. Non-linear navigation aids are usually provided through tables of contents, lists of figures and keyword or subject indices. Alternative exploration trails are often specified through the judicious use of cross-referencing information, "see also" listings, or even by explicitly instructing the reader to follow specific paths depending on their knowledge or interests. In short, the claim here is that *it is not the abstract form of the printed document which is linear but rather the medium in which it is delivered ie paper.*

A conventional document form can provide a strong and familiar structuring metaphor with which an author can organise their *hypertext* material. But even more importantly the same metaphor will be shared by potential readers, and this can help reduce any "disorientation" feelings [2]. These observations can help account for the fact so many existing HTML documents are organised in this traditional way.

3.1.3 It is difficult to determine the level of granularity for each hypertext node

This is also known as the "fragmentation problem" [1]. Any process which automatically decides on the amount of information that should go into each node cannot be guaranteed to always produce good results. Too fine node granularity may interrupt the way in which an idea is presented while too coarse granularity may result in unrelated ideas being grouped together in the same hypertext node.

Assuming an underlying hierarchical structure, one strategy may be to fragment a document on the basis of sectioning information (chapters, sections, subsections etc) but also *allow the author to override the default level of granularity.* This is possible by converting a document into a single HTML node, or a small set of "coarse" nodes, or a larger set, depending on a specified "depth" at which fragmentation should stop. See the example on structure extraction (Section 4) to see how this can be accomplished.

3.1.4 Some documents require custom navigation aids

One possible strategy with respect to this problem is to try to automate the generation of navigation aids based on the underlying document structure but also allow the author to customise or extend them when that is desirable.

With LaTeX2`HTML` a "navigation panel" is placed automatically at the top of each generated page (and the bottom of pages with more than a specified number of words). The default contents of this navigation panel can be changed by the user. Also, the author is free to construct alternative navigation aids manually. An example of this is shown in the map of Figure 1 where in the online version of this document any node can be accessed by clicking on its graphical representation in the map.

3.1.5 The conversion process is inflexible and converted documents cannot take full advantage of hypermedia capabilities

One of the great strengths of hypertext links in HTML is that they can point to arbitrary external locations or services. Also, other features such as fill-out forms or active image maps would seem to be denied to those that choose the automatic conversion route.

There are two ways one may deal with this problem. An obvious solution would be to "post-process" the output from the conversion process in order to add external links or fill-out forms. This however can lead to maintenance problems when the source document needs to be updated.

Another approach is to extend the source word-processing format with new commands or macros. These could be designed so that they appear in a sensible way in the paper version of the document (eg external links could appear as footnotes). But when processed by the conversion system these could become active external links, embedded images or forms. For those that want maximum flexibility a "catch-all" macro could be available so that raw HTML tags could be embedded directly in the source document.

This mechanism is already available in LaTeX2`HTML` where it has been put into good use to provide active control panels in online workbooks [7], active bibliographic references (eg at [12] or as in this document), or alternative navigation aids (like the active map in Section 1 of the online version of this document.

3.1.6 There are subtle differences between electronic and paper-based versions and this will compromise quality

Obtaining an electronic hypertext version and a paper version from the same document source is likely to compromise the quality of at least one of them. Often this is due to subtle differences in the wording or the amount of information that is necessary to give to the reader which may be different depending on the delivery medium. For example the electronic version may contain instructions on how to fill out and submit an online form, while the paper version may contain instructions on how to submit a form by fax or traditional mail.

Another example of such differences may be the way in which cross-references are constructed in paper-based documents usually with "numerical indirection" eg

> Conditional text (see Section 3.1.6) is text which may or may not appear in a document depending on the delivery medium

In a hypertext document the above sentence could be expressed much more succinctly as follows:

> Conditional text is text which may or may not appear in a document depending on the delivery medium

Of course this problem can be dealt with using manual post-processing but again this may cause future maintenance problems. An alternative is "conditional text" as defined in the example above where the author specifies which parts of a document are intended only for electronic delivery and which should only appear in the paper version. This can be done with commands or macros which are interpreted differently by the word-processing system and by the conversion system. Subtle differences between the paper and electronic versions of this paper were introduced using this technique.

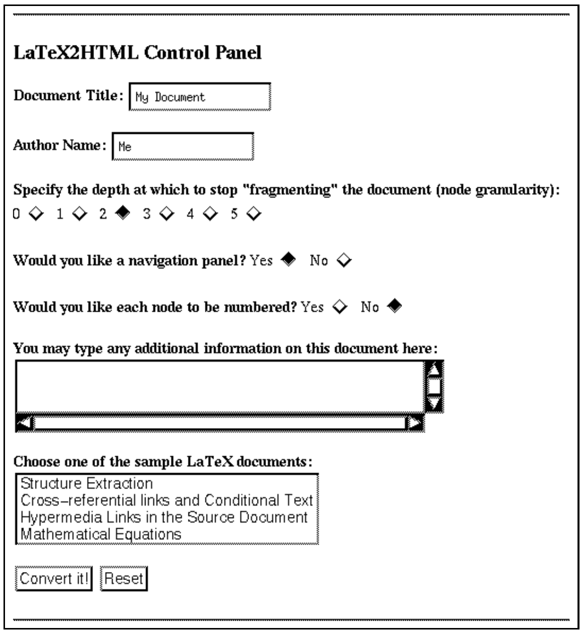
4 A Case Study in Text to HyperText Conversion: LaTeX2`HTML`

LaTeX is a "document preparation system" based on Donald Knuth's TeX typesetting program [6]. It is really a set of TeX macros that automate the formatting of different types of documents. Despite the current lack of freely available wysiwyg editing tools, LaTeX owes much of its popularity to its fine control on page layout, its extensibility through external TeX style files and its excellent facilities for working with mathematical equations.

LaTeX2`HTML` [5, 4, 13] is a text to hypertext conversion system that can translate a LaTeX source document into a "web" of HTML files. It is being widely used[2] in a variety of contexts some of which are listed below.

- Electronic books such as the CRS4 Active Books Library[3] and the Computational Science Education Project[4]
- Scientific papers such as those on the MIT transit project[5], a document on the WorldWide Web[6] (with French accents) and one on electronic submissions to an IEEE journal[7]
- Lecture notes, supporting documentation and coursework[8]
- Online training material[9], system documentation[10] and user manuals[11]

The way in which LaTeX2`HTML` handles the conversion from text to hypertext can best be illustrated with a set of interactive examples. In the online version of this document, readers can make use of the control panel (fill-out form) shown below. They can select one of the predefined examples described in the following sections and submit them for conversion to an experimental LaTeX2`HTML` service set up at the University of Leeds. The results from the conversion according to the readers' selections are then displayed on the screen after a short delay. In this non-interactive version only some sample screen snaps have been included.



LaTeX2HTML Control Panel

Document Title: My Document

Author Name: Me

Specify the depth at which to stop "fragmenting" the document (node granularity):
0 ○ 1 ○ 2 ● 3 ○ 4 ○ 5 ○

Would you like a navigation panel? Yes ● No ○

Would you like each node to be numbered? Yes ○ No ●

You may type any additional information on this document here:

Choose one of the sample LaTeX documents:
Structure Extraction
Cross-referential links and Conditional Text
Hypermedia Links in the Source Document
Mathematical Equations

---

[2] Well over 1200 copies of the current version have been retrieved since January 1994.

[3] http://www.crs4.it/HTML/int_book/meta_page.html

[4] http://compsci.cas.vanderbilt.edu/csep.html

[5] http://www.ai.mit.edu/projects/transit/tn-cat.html

[6] http://web.urec.fr/docs/WWW/WWW.html

[7] http://www.research.att.com/esubmit/esubmit.html

[8] http://www.cm.cf.ac.uk/lecture_notes.html

[9] http://www.strath.ac.uk/CC/Courses/OnlineTraining.html

[10] http://www.cwi.nl/cwi/people/Guido.van.Rossum/python-tut/tut.html

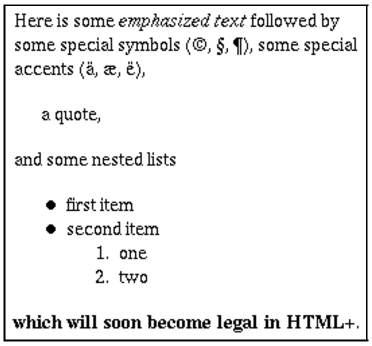
[11] http://olt.et.tudelft.nl/usr1/patrick/public_html/docs/wwman/wwman.html

The control panel of the Leeds LaTeX2HTML test service.

Example 1: Basic Presentation

The LaTeX source code below shows some of the presentation commands.

```
Here is some {\em emphasized text}
followed by some special symbols
(\copyright, \S, \P),
\input{german} some special accents ("a, \ae, "e),
\begin{quote}
a quote,
\end{quote}
and some nested lists
\begin{itemize}
   \item first item \item second item
   \begin{enumerate}
      \item one \item two
   \end{enumerate}
\end{itemize}
{\bf which will soon become legal in HTML+}.
```



Here is some *emphasized text* followed by some special symbols (©, §, ¶), some special accents (ä, æ, ë),

a quote,

and some nested lists

- first item
- second item
  1. one
  2. two

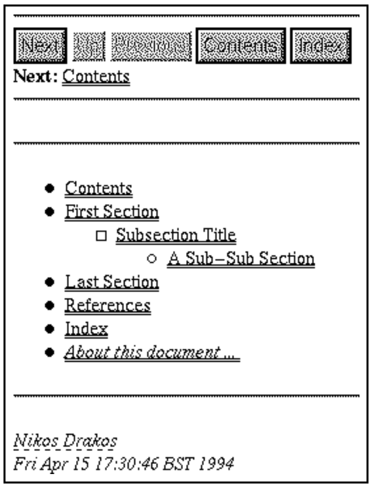
which will soon become legal in HTML+.

The result from converting Example 1, viewed with Mosaic for X.

Example 2: Structure Extraction

The basic structure of a LaTeX document is expressed using "sectioning commands" as shown below.

```
\tableofcontents
\section{First Section}
This is the first section.
\subsection{Subsection Title}
And some text with a bibliographic
citation \cite{nobody}.
\subsubsection{A Sub-Sub Section}
Some text with an index entry
\index{index entries}.
\section{Last Section}
\begin{thebibliography}
\bibitem{nobody} A Nobody.
\newblock {\em User's Guide \&
Reference Manual}.
\newblock GlobeWide Publishing
Company, Inc., 1995.
\end{thebibliography}
\begin{theindex}
\end{theindex}
```



Next: Contents

- Contents
- First Section
  - Subsection Title
    - A Sub-Sub Section
- Last Section
- References
- Index
- *About this document ...*

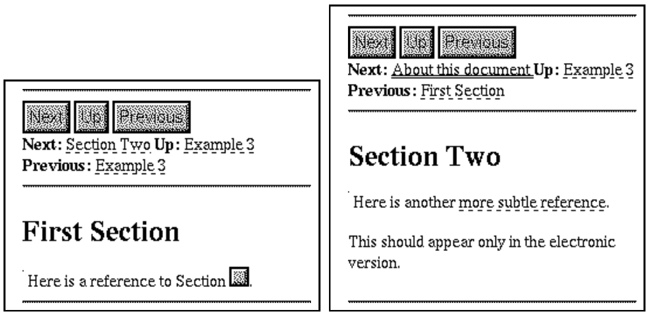
Nikos Drakos
Fri Apr 15 17:30:46 BST 1994

One possible outcome based on structure extraction (Example 2) showing a generated navigation panel and structural links.

Example 3: Cross-referential links and Conditional Text

Cross-references in a LaTeX document are expressed as pairs of `label-ref` commands as shown below[12].

```
\section{First Section}
\label{one} Here is a reference
to Section \ref{two}.
\section{Section Two}
\label{two} Here is another
\hyperref{more subtle reference}
{reference (Section }{)}{one}.
\begin{latexonly}
This should appear only in
the paper version.
\end{latexonly}
\begin{htmlonly}
This should appear only
in the electronic
version.
\end{htmlonly}
```



Next: Section Two **Up:** Example 3
Previous: Example 3

First Section

Here is a reference to Section ▨.

Next: About this document **Up:** Example 3
Previous: First Section

Section Two

Here is another more subtle reference.

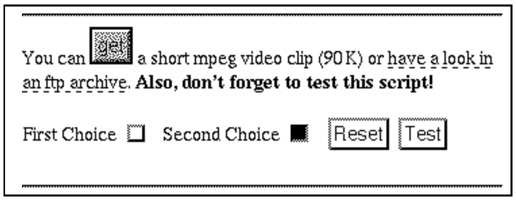
This should appear only in the electronic version.

Two of the nodes resulting from the conversion of Example 3.

Example 4: Hypermedia Links in the Source Document

The example below shows some new commands defined using the LaTeX macro language for specifying external links, inlined images and raw HTML tags.

```
\hrule
You can \htmladdnormallink{
\htmladdimg{http://cbl.leeds.ac.uk/nikos/figs/get-motif.gif}}
{http://cbl.leeds.ac.uk/nikos/movies/jet.mpeg}
a short mpeg video clip (90K) or  \htmladdnormallink{have a look in an ftp archive}
{ftp://ftp.tex.ac.uk/pub/archive/support/latex2html}.
\begin{rawhtml}
<B>Also, don't forget to test this script!</B>
<FORM METHOD="POST" ACTION="http://cbl.leeds.ac.uk/nikos-cgi/test-cgi">
First Choice <INPUT TYPE="checkbox" NAME="a" VALUE="1">
Second Choice <INPUT TYPE="checkbox" NAME="b" VALUE="2" CHECKED>
<INPUT TYPE="reset" VALUE="Reset">
<INPUT TYPE="submit" VALUE="Test">
</FORM>
\end{rawhtml}
\hrule
```



You can ▨ a short mpeg video clip (90K) or have a look in an ftp archive. **Also, don't forget to test this script!**

First Choice ☐ Second Choice ☒

The result from converting Example 4.

Example 5: Mathematical Equations

LaTeX has excellent support for mathematical equations through predefined mathematical symbols and commands that control layout, as well as automatic numbering justification. Also, generated

---

[12] A more extensive treatment of the cross-referencing mechanism and how it has been extended to apply not just within documents but also *between documents* can be found in the LaTeX2HTML manual [5].

equation numbers can be used as cross-references in the main text. A small example is shown below.

```
\section{First Section}
Here is the first equation
\begin{equation}
\left[ {d^2 \over dt^2} + m^2 \right] \phi_{\rm cl}(t) + {1\over3!}
\lambda \phi_{\rm cl}^3(t) = 0 \,, \label{diffeq}
\end{equation}
and another
\begin{equation}
\langle 0 {+} | \phi(x) | 0 {-} \rangle^{\rho} \to \phi_{\rm cl}(x) \,,
\end{equation}
an inlined one $\phi_{\rm cl}(t)$, and some special
symbols $\alpha \delta \aleph \lambda$.
\section{Second Section}
Here is a link to the first equation (\ref{diffeq}).
```