HyperTeX: a working standard

Arthur P. Smith (asmith@mammoth.chem.washington.edu)
Dept. of Chemistry, BG-10, University of Washington, Seattle, WA 98195

The popularity of the World-Wide-Web and the HTML hypertext language for electronic distribution of information over the internet has thus far been difficult to extend to scientific documents, in part because TeX, the word processing language of choice for technical and mathematical documents, does not have the hypertext capabilities necessary for full integration into the WWW. HyperTeX consists of a collection of extensions of TeX to handle both internal and external (local or networked) hypertext, thus allowing the huge body of TeX-based documents to be more fully integrated with the WWW.

Contents

I. INTRODUCTION

The past year has seen a revolution in the processes of internet-based information navigation and retrieval with the advent of easy-to-use graphical browsers (in particular Mosaic) based on the World-Wide-Web (WWW). The revolution is a result of two components - first the browsers allow a near-uniform (point-and-click or other method) access to documents in almost any format (interpreted according to local .mailcap files) and from almost any internet-based source, accessed as regular files or via ftp, gopher, http or one of many other possible methods, and along with this the Universal Resource Locator (URL) mechanism provides a surprisingly easy and uniform way to specify the location of any document on the net. Second, for certain classes of documents (html files, or gopher text files) embedded URL's or other addresses are understood to refer to other, external, documents which can be followed according to the interests of the person viewing the document, producing an interconnected web of documents.

The goal of the HyperTeX collaboration is to extend this second privileged class of documents to include documents based on TeX , the word-processing language of choice for mathematical and scientific writing, thus fully incorporating TeX documents into the burgeoning **web** of information on the internet.

II. WHY HYPERTEX?

There already exists one approach for incorporating TeX documents more fully into the **web** - conversion to HTML, as in the program latex2html by Nikos Drakos. This can work very well, and is already used in some of the electronic publications in mathematics, but there are also several serious problems with this, aside from the technical issues associated with the complexity of the conversion process. HTML by design allows very little author control of the visual form of a document. This is touted as an advantage because it preserves only the *essential* elements of a document and not the artificialities of a page - in fact HTML documents do not have pages at all, although some of the sense of a *page* is implied by separation of a single document into many files. Aside from loss of author control, there is a practical problem of a lack of mathematical tools in the current implementations of HTML - tables and equations are either difficult to implement or impossible. Latex2html gets around this by conversion of such things to bitmapped images, but this is an inefficient and expensive process - and goes in just the opposite direction of HTML's theme of extracting the *essence* of a document, making the document essentially unreadable without a good network connection and a computer with a high quality display.

These problems with HTML are compounded if scientific authors attempt to write documents directly in HTML rather than using TeX first - the lack of authoring tools, the absence of macro capabilities, and the ill-defined nature of the language make this an unpleasant task - just dealing with ordinary text is easy, but getting Greek letters, mathematical symbols, equations and tables into your document is not. The one nice feature of HTML is the ease with which figures can be incorporated into a document. But at least PostScript figures can be incorporated into a TeX document with equal ease using modern dvi interpreters, and the HyperTeX standard presented here allows arbitrary images and other external documents to be referred to and brought to the screen with a single mouse click.

The point of all this is that hypertext capabilities, and the use of URL's to locate new documents - the main feature of HTML that makes it such a useful network information navigation tool - can be much more easily incorporated into TeX than the mathematical capabilities of TeX and the years of experience embedded in various TeX macro packages can be incorporated into HTML. Whether TeX in general provides a better model for the viewing of on-line information remains to be seen.

III. HOW DOES IT WORK?

The underlying element of our implementation of HyperTeX is the use of a TeX macro that bypasses the TeX interpretation process and sends a message directly to the dvi interpreter that processes TeX output. This is the **\special** macro, previously used to define procedures for drawing or including figures in TeX documents. When the characters **\special{string}** appear in the TeX document, the *string* is passed directly without interpretation to the output dvi file (preceded by a marker to identify this as a *special* message to the dvi interpreter). The dvi previewers or processers then interpret this string according to its first few characters. The original HyperTeX specification (due to Paul Ginsparg, Tanmoy Bhattacharya, and me) uses the initial characters *html:* to denote HyperTeX elements in an HTML-like style. David Oliver (oliver@gang.umass.edu) has introduced a slightly different specification that uses the initial characters *hyp* to denote his own style of HyperTeX. I will discuss only the original specification in this paper, since as far as they are currently implemented both specifications are essentially equivalent. Note that dvi interpreters that do not understand the *html:* or *hyp* special commands will ignore them, or at worst print out warning messages. Therefore dvi files processed to include HyperTeX commands are fully compatible with old dvi interpreters.

After the initial *html:* string, the specification is identical to a restricted form of HTML. The five arguments we have added to the **\special** command are:

href: `\special{html:`}

name: html:

end: html:

image: html:<img src = "href_string">

base_name: html:<base href = "href_string">.

The *href*, *name* and *end* commands are used to do the basic hypertext operations of establishing links between sections of documents. The *image* command is intended (as with current html viewers) to eventually place an image of arbitrary graphical format on the page in the current location. Currently for xhdvi, *image* brings up an external viewer with the image, if such a viewer is available. The *base_name* command should be used to communicate to the dvi viewer the full (URL) location of the current document so that files specified by relative URL's may be retrieved correctly.

The href and name commands must be paired with an end command later in the TeX file - the TeX commands between the two ends of a pair form an *anchor* in the document. In the case of an href command, the *anchor* is to be highlighted in the dvi viewer, and when clicked on will cause the scene to shift to the destination specified by *href_string*. The *anchor* associated with a name command represents a possible location to which other hypertext links may refer, either as local references (of the form **href="#name_string"** with the *name_string* identical to the one in the name command) or as part of a URL (of the form *URL#name_string*). Here *href_string* is a valid URL or local identifier, while name_string could be any string at all: the only caveat is that "' characters should be escaped with a backslash ('`\`'), and if it looks like a URL name it may cause problems. There may also be problems if LaTeX tries to interpret the *href_string* or *name_string* - in that case preceding the command with **\protect** should usually work. Any defined *name_string* can be referred to in any href referring to the document, in the form **href="URL#name_string"**. Note that anchors may be nested. The only restriction in current implementations is that anchors are truncated at page boundaries.

Because this html-based naming scheme is somewhat unwieldy, although very general, Tanmoy Bhattacharya (tanmoy@qcd.lanl.gov) has written several collections of TeX macros to simplify things. The basic package is *hyperbasics.tex* which defines the following simple low level hypertex macros:

- `\href{url}{text}`: text becomes an href anchor referring to *url*.
- `\hname{myname}{text}`: text becomes a name anchor with name *myname*.

plus others that are used to automatically convert LaTeX or other style markup into corresponding names and references.

IV. HOW DO I USE IT?

A. As a reader

There are currently two dvi interpreters that understand the HyperTeX `\specials`: xhdvi for X windows, and HyperTeXView.app for NextStep. We are proceeding with work on a dvi-pdf converter that understands

HyperTEX , and we are encouraging work on dvi previewers or TEX authoring tools for Macintosh and PC that incorporate HyperTEX elements.

For a TEX document that has already been processed to a dvi file with HyperTEX elements, viewing the internal hypertext is almost trivial - you just fire up the dvi previewer and navigate by button clicks as with Mosaic or other WWW browsers. To have xhdvi, for example, brought up automatically from Mosaic when a dvi document is referenced, you need to have a *.mailcap* file in your home directory, and create or modify the line:

application/x-dvi; xhdvi

Your machine must already have the TEX essentials on board of course - in particular the pk font files, and the location of those font files needs to be communicated to the previewer. If xdvi is already working for you, xhdvi should work too. Details for getting xhdvi working on your machine are provided below.

For jumping to external documents from within the hypertexted dvi file, a couple of additional elements are needed, also desribed below for the case of xhdvi.

B. As an author

Here is where the power of TEX's macro capabilities appears. A working internal hypertext document can be made from a LATEX document with a one-line addition to the file, using Tanmoy Bhattacharya's hypertex macros. These macros convert the standard LATEX markup into hypertext links between the different sections of the document, so that references to equations, tables, footnotes, and section headings are in place, and bibliographic references and figures refer back at least to the bibliography entry or figure caption. These in turn may be set to refer to corresponding external documents but this process is not automatic - currently the author will have to add these references by hand, although automatic procedures can be envisioned. With an internet connection, xhdvi can be used to preview the document and check that the references actually work, before the document is submitted to the archives.

The macros developed thus far use standard naming conventions for the underlying structures in LATEX and other standard macro packages, so that appending #equation.2.3, #page.7, #figure.4, #table.2, etc. to the URL for any TEX file processed with these packages will go to the right place, allowing easy hypertext reference to the internal structure of other documents.

In order to get started, however, you need to place these macro files in one of the standard areas that your TEX looks for input files (you can modify your TEXINPUTS environment variable to get it to look in your own directories). The needed macro files are itemized in the HyperTEX introductory document at http://xxx.lanl.gov/hypertex/index.html and can be obtained in one lump by anonymous ftp from ftp://snorri.chem.washington.edu/hypertex/hypermacros.Z.

C. As an e-print manager

Since we currently only have dvi previewers, an e-print server would have to serve the documents in pre-processed *.dvi* form. This means converting documents to HyperTEX if the author has not already done this, and possibly applying automated insertion of URL's corresponding to references in the bibliographic section. The manager could do this by hand but it might be rather time-consuming.

For ease of use, the best way to serve the documents is probably as a combined package of .dvi and .ps files that go together. This requires the e-print manager to create a new content-type associated with this package, and to supply an unpackaging program for the reader to place in their *mailcap* file, which automatically calls up *xhdvi* or another HyperTEX browser on the resultant main dvi file. The reason for doing this is that .ps files included by the standard *figures in TEX* macros will not generally be understood as remote documents, at least at the current level of previewer capabilities. Another option in this unpackaging method is to supply the TEX file itself, pipe it through a simple converter to HyperTEX and through TEX itself, and then call one of the HyperTEX viewers. These approaches are already in use at some locations (e.g. CERN).

When the pdf converter is available, the entire document should come as a single pdf file, unless the document refers to non-PostScript images or other inclusions in which case the packaging approach (or use of absolute URL's) remains necessary.

V. HOW DO I GET IT?

Currently the following are available:

xhdvi version 0.6 at ftp://snorri.chem.washington.edu/hypertex/ - A HyperTEX viewer based on xdvi-18, modified by Arthur Smith. Precompiled versions for various UNIX architectures are available in the same directory.

HyperTeXview.app courtesy of Dmitri Linde (dmitri@physics.stanford.edu [also the author of InstantTeX.app]) for NextStep, precompiled for Motorola and Intel-based NeXT machines. See http://xxx.lanl.gov/hypertex/index.html for availability.

The macro and style files listed above by Tanmoy Bhattacharya, available at ftp://nqcd.lanl.gov/people/tanmoy/hypertex/

VI. DETAILS ON XHDVI

Xhdvi retains all the features of the latest version of xdvi (version 18) and adopts in addition many of the hypertext features of Mosaic, the most popular WWW browser. Hypertext links are underlined or altered in color (the underlining can be turned off) and a left-mouse click on a link causes the view to shift to the destination point for the link, as long as the destination is another dvi file. If the link is not to a dvi file, an external viewer is employed, following the mime and mailcap definitions or using standard defaults if those are not locally defined. A middle mouse click on a link brings up a new viewer whether or not the destination is a dvi file - this is intended to be useful to refer back to equations or to bring up footnotes, since the new dvi window is small. There are also a large number of keyboard accelerators, all described in detail in the man page.

In general, see the installation notes provided with xhdvi.

In outline what is needed is:

1. The compiled xhdvi program - precompiled binaries are available for Sun, NeXT, SGI, HP, IBM RS6000, or you can get the source and compile it yourself. Let me know of any compilation troubles - it's written in C.

2. The TEX fonts, at least in pk format. If xdvi, dvips or some other dvi interpreter are working on your machine then they must be around somewhere. Otherwise you can get a small sample set of fonts from snorri.chem.washington.edu, or get the full TEX distribution from somewhere...

3. If the TEX fonts are not in the directory /usr/local/lib/tex/fonts you will either have to recompile xhdvi or use the environment variables, or you can add a symbolic link from /usr/local/lib/text/fonts to the right places. On a machine where you don't have superuser access and you have to install the TEX fonts yourself (a few Megabytes) you could put them in ~/TEX/fonts and then set the environment variables:

   setenv XDVIFONTS ~/TEX/fonts/pk

   setenv XDVIVFS ~/TEX/fonts/vf

4. Set up the connections between the Web browser and xhdvi. If you use mosaic for example, setenv WWWBROWSER /usr/local/bin/mosaic will let xhdvi know what to send HTML files to. To let mosaic know to bring up xhdvi for any dvi files, you need to have the line

   application/x-dvi; xhdvi

   in your ~/.mailcap file.

5. The application defaults file XHdvi should be installed in the standard application defaults directory on your machine, or you can take lines from it and modify them for your own taste and put them in your ~/.Xdefaults file. For example I use the following resource specifications to get a particular size and position of the window with white on black lettering and with the hyperlinks in cyan, and to remove the buttons:

   xhdvi*geometry: 800x600-0-0

   xhdvi*foreground: white

   xhdvi*background: black

   xhdvi*highlight: cyan

   xhdvi*expert: true

6. You need to have the ghostscript program on your machine and in your default execution path in order to view postscript from xhdvi. Similarly, other viewers defined in the mailcap file should be available on the machine.

7. You need to install the man page xhdvi.man in /usr/local/man/man1 and add /usr/local/man to your MANPATH environment variable in order for *help* to work from xhdvi.

VII. SOME EXAMPLES

This document is available in raw HyperTEX format and in converted dvi format *via* anonymous ftp at the address ftp://snorri.chem.washington.edu/hypertex/asmith.tex. The HyperTEX version of this paper uses the two-column APS journal style of revtex. The table of contents at the beginning is generated automatically with the LATEX **\tableofcontents** command.

See also the examples provided by Paul Ginsparg in the HyperTEX introductory document at http://xxx.lanl.gov/hypertex/index.html. Some of these are files randomly selected from the HEP archive, including LATEX, RevTEX, and other formats.

VIII. WHAT STILL NEEDS TO BE DONE?

Unfortunately, at this point reference to networked files (via URL's) suffers from a couple of problems. Xhdvi does not yet include any of the network transport code that ordinary WWW browsers use, and the intention was to avoid having to add this layer of complexity by communications back and forth with a WWW browser. However, such communication is as yet not standardized, and suffers from its own problems. So currently, when Xhdvi comes across a URL reference, it forwards it directly to the WWW browser (defined by environment or Xresource variables) so that a reference to an external dvi file would bring up a new instance of the WWW browser which would in turn bring up a new xhdvi viewer. This is a rather inelegant solution, but it is perhaps sufficient at the moment. A better solution will come along, and it may simply be inclusion of network transport code in the xhdvi viewer itself, to make it a competing WWW browser...

The other problem is that if brought up by a WWW browser, xhdvi is not provided with the absolute URL information used in obtaining the dvi file it is working on, and so cannot pass this information on to further instances. Therefore, relative URL's in a HyperTEX document (unless they can be guaranteed to be to local files that would have been transported along with the dvi file) will not work.

Both of the above are problems intrinsic to current WWW browsers, and we are working on promulgating solutions to these.

IX. HOW DO I STAY IN CONTACT?

The Hypertex discussion group is a mailing list based at snorri.chem.washington.edu and maintained by myself. Send me e-mail (asmith@mammoth.chem.washington.edu) if you want to join the list, or send queries directly to the mailing list:

hypertex@snorri.chem.washington.edu