Putting the Hyper back in Hypertext

Bill Wadge
m.c. schraefel

Computer Science Department
University of Victoria

Summary: we describe a set of front-end macros for IHTML which make it extremely easy to produce "real" hypertexts (Conversational Texts) which not only come much closer to Ted Nelson's original concept than is practical using ordinary HTML, but refine that concept as well.

Two Cheers for the World Wide Web
The World Wide Web may be one of the technological wonders of the world, but there's still plenty of room for improvement.

The Web is based on the notion of hypertext, a concept invented and named by Ted Nelson in the [Nelson 1987]). In many people's minds, hypertext is the Web - a collection of pages with embedded links to other pages.

Nelson, however, proposed a much more general and open-ended definition of hypertext. He described it simple as nonsequential writing, in which the reader "can go instantly in a choice of directions from any given point".

In traditional, paper-based sequential writing the reader proceeds from beginning to end, and from the top of each page to the bottom. Some forms of print depart slightly from sequentiality. The pages of newspapers and magazines present the reader with a mosaic of articles and images. Much of the information is displayed in charts and graphs, or in collections of discrete points not in any logical order.

The individual articles, however, are entirely sequential, and the magazine or book as a whole is intended to be read in a more or less from front to back. These limitations are imposed by the physical properties of printed paper as a medium.

The Web, in spite of its enormous success, is based on a very conservative notion of hypertext. The basic unit is the page, and the only nonlinear feature is the link, which takes the reader from one page to the next. At any given point the readers have only a very limited set of directions in which to proceed. They can
scan the visible part of the current page; proceed linearly (scroll) through the current page; or jump to one of the pages linked from the current one.

In Nelson's words, the Web is "the minimal concession to hypertext that a sequence-and-hierarchy chauvinist could possibly make". It is hardly surprising, then, that even (or especially) the best-designed sites copy the magazine format: a small collection of pages, each a mosaic of linear text. Most could be printed and browsed off-line, on paper. The real wonder of the Web is the large number of magazine-sites that are available, and the astounding speed with which this collection can be searched.

Intensional HTML

The Intensional HTML (IHTML) project [Wadge et al, 1998] is an attempt to repair the deficiencies in HTML by making it practical to author Web pages that vary over a multidimensional version space. (This work grew out of the system of [Plaice and Wadge, 1993] originally designed for software version management).

At first sight IHTML addresses only one of the deficiencies of HTML (lack of versioning), at first sight not the most serious. However, as schraefel showed in her dissertation [schraefel 1997], multidimensional versioning makes possible the authoring of parametrizable documents that the reader can interactively reconfigure. Each transaction with such a ConText (Conversational Text) is a choice made by the reader as to the direction of the next step to take in the process of browsing, and to some extent in creating, the document they will read.

The result is that documents designed with the ConText paradigm can offer the reader a much richer choice of possible documents to render/read based on the user's needs for that text. For instance, depending on the user's dimensional choices, they have many more ways of exploring the material presented, such as looking at smaller, more summarized versions of a document. This latter point stems from schraefel's ConText paradigm of versioning on "Degree of Detail", a more refined version of what Nelson refers to as "stretchtext"

Degree of Detail in IHTML

Nelson's Stretchtext is text that exists in many different lengths and whose length can be varied interactively by the reader. Nelson suggested a length control lever; pulling back on the lever causes new words to appear in previously empty spaces, while pushing forward has the effect of removing them again. With the lever pushed all the way forward, we might see only the title; with it pulled all the way back, we might get the full text of a long article.
In 1997, Schraefel envisioned this notion of stretching text as expos- ing variable degrees of detail in a document. These degrees of detail can also be versioned across other dimensions such as Level of Expertise of any given document version.

As we shall see, the notion of degree of detail is part of a paradigm for document construction that allows the reader to "focus" on parts of a document. In ConTexts, this focus would be applicable to a document on both global and local levels. That is, Degree of Detail switches would be optionally applied both to the entire document and/or to an individual document component. This global/local versioning is a further refinement of the stretchtext paradigm.

We can formalize a stretchtext as a ConText that varies in a single numerical depth parameter (dimension). At depth 0 we see only the title, at depth (say) 10 we see the whole article.

This kind of variable-depth document can easily be implemented using the HTML \texttt{<i/select>} construct (a kind of case statement) to conditionally include portions of text.

Suppose, for example, we want the depth 0 text to read

\begin{quote}
Computer screens will be the entire structure of tomorrow's publishing and libraries.
\end{quote}

and we want the depth 1 text to read

\begin{quote}
Computer screens will be not merely the basis of office systems and systems for writers but eventually the entire structure of tomorrow's publishing and libraries.
\end{quote}

The following HTML fragment

\begin{verbatim}
<isect>
  <i.case version=depth:1>
    not merely the basis of office systems and systems for writers
    but eventually
  </i.case>
</isect>
\end{verbatim}
the entire structure of tomorrow's publishing and libraries.

will evaluate to the first in a context in which the coordinate in the depth dimension is 0, and to the context in which the depth dimension is 1 (or greater).

If in addition we want the word newfangled to appear just before systems for writers, we use two levels of nesting:

```
<iselect>
 <icase version=depth:1>
  not merely the basis of office systems and
  <iselect>
   <icase version=depth:2>
    newfangled
   </icase>
  </iselect>
 systems for writers
 but eventually
 </icase>
</iselect>
```

the entire structure of tomorrow's publishing and libraries.

**Stretchtext for the Millions**

This implementation technique is simple in principle but not ideal (when applied directly) in practice, for two reasons. First, the author must add four lines of markup for every piece of text that can appear and disappear. These four lines are not all the same for each segment, and are not necessarily contiguous.

The second, more fundamental problem is how to author the text in the first place (this applies to all forms of hypertext). Schraefel has suggested generating collections of "chunks" (e.g. aphorisms) which then are selected and collected in various ways according to the current context.

We describe here a definitive solution to the technical problem of dealing with legacy documents, and a provisional solution to the authoring problem for ConTexts.

The technical solution is relatively straightforward, namely to provide a preprocessing tool which translates a simple higher level notation. The exact form taken by the higher level notation is of vital importance. After some experimenting, Wadge settled on well-tested pre-existing system: troff macros!
Troff is an early typesetting program still supplied with UNIX. Troff has been largely superseded by TeX, which in turn may soon give way to MathML. Troff, like TeX, has a built-in macro processor. By good luck (and good design) the troff macro evaluator can be used entirely separately from the formatting and typesetting features. All one has to do is turn off filling (with the .nf command) and invoke the -a option for an ASCII approximation.

For stretchtext, we need exactly two macros, say DB and DE.

A call to DB signals entry into a section of text that stays hidden until the depth is one more than that needed to make the surrounding text appear; DE signals the end of such a section. The stretchtext given above can be written

```
Computer screens will be
.DB
not merely the basis of office systems and
.DB
newfangled
.DE
systems for writers
but eventually
.DE
the entire structure of tomorrow's publishing and
libraries.
```

The definitions of these two macros are straightforward, although they use an absolutely vital feature of the troff processor, global state (in the form of number and string registers).

```
.DE DB
<i/select>
<i/case version=depth:\\n+d>
.. 
.DE DE
.nr d \n-d
</i/case>
</i/select>
.. 
```

The provisional authoring methodology should now be obvious: take existing, legacy text (or HTML) and add macro calls to parenthesize less and less important sections of the text. There is no a priori reason why this should work at
all; no logical reason to expect that we can remove large sections of an existing
text and have something sensible left over. In practice, however, it works very
well. Possibly, because one property of well-written text is that the main points
are stated somewhere clearly and concisely.

Poptext

We can use this same idea to implement Degree of Detail (which we here
rename as the more concise PopText).

Poptext is like stretchtext in that we can choose to reveal or hide parts of the text
as we currently see it. It differs in two respects. First, each individual piece of text
can be controlled both locally and (in its more elaborate version) globally. A text
component can be expanded and contracted locally (made to pop in or out of
view on its own) to its own degree of detail, or all text components on a page
can be made to expand or contract to the same degree of detail globally.
Second, the poppable pieces at a given level do not disappear altogether --
typically, a title, heading, or additional text remains visible.

Poptext can dramatically simplify the appearance of a text, by hiding details and
allowing the reader to focus on particular parts of the document. For example, a
poptext version of this paper might initially look like

>Summary
>Two Cheers for the World Wide Web
>Intensional HTML
>Stretchtext
>Stretchtext for the Millions
>Poptext

Clicking on an arrow to the left of a title causes the body to appear, displacing the
text below it downwards. Poptext can be nested; clicking on the Stretchtext for
the Millions arrow may not bring (say) the macro definitions immediately into
view.

For most purposes we can convert legacy HTML to poptext using three macros:
SC, SD and SE. SC begins a poptext section; SD marks the beginning of the text
that pops in and out; and SE marks the end of the section (the text between SC
and SD is, typically, the heading that remains in view). (Currently these macros
allow only local popping and unpopping).

Here are the definitions for a basic form of poptext (it uses a long narrow table to
indent the popped section).
The notion of expanding and contracting text alone is hardly a new idea; many word/outline processors allow the author to expand or collapse sections and subsections. The same idea appears in the Mac OS list view of directories. There are even Web sites (very few) which offer such outlining views, implemented using DHTML or (as at www.interval.com) cgi scripts.

However, deploying Poptexts within the ConText paradigm, two things become possible. First, it is possible to take even large legacy documents and convert them into user-controlled versioned views in literally minutes. Furthermore, with the IHTML approach we can mix features like degree of detail and level of expertise (as explored in schraefel, 97).

Choptext and Beyond

It is not hard to imagine many useful variations on the constructs like poptext. We could allow the user to specify a brief summary text to be displayed when the poptext is not visible. We could arrange that at most one section is popped, so that popping another section folds the one that is currently open (this avoids making the document too long). A more refined arrangement would allow only
one open at each level, so that the user can drill down to subsections and sub-subsections but still read them in context. Also, we can combine stretchtext and poptext: we could have a global control on the length of the texts being popped. These possibilities are relatively easy to program in IHTML.

One of the more interesting ideas is allowing links which make text disappear completely (we call this choptext). For example, we could have an X icon next to the arrow icon for each section. Clicking the X (i.e. following the link anchored to it) removes all trace of the entire section. In other words, the X links us to a version of the page in which the folding text, the summary text, the arrow icon and the X itself have all vanished.

This would be useful, for example, in preparing a lecture. If the lecture notes were already on-line, we could click away those sections that we want to skip in the lecture. In fact, if we use video projection, we can erase them in real time - just as we can remove notes for topics we have finished covering.

Once we have the ability to make text selectively appear, expand, shrink, and disappear, we have also begun to erase the boundaries between documents themselves. For example, some of the chopable popsections might expand to a whole chapter of the textbook. The lecturer could, if they wanted chop out all the other sections and spend the time going over the text itself.

In other words, we can incrementally transform our own lecture notes into a section of the text. Of course we can already follow hypertext links, but that takes us from one document to another in a series of abrupt leaps. With poptext and choptext, we can morph one document into another in small steps, in which each new piece of information appears in a context that explains its relevance.

And we can do this without preparing, beforehand, a huge file incorporating both our notes and the text. Instead, we can put IHTML include's in popsections; they will be expanded only when that part of the iselect is actually selected.

It should also be clear that in so doing we have almost completely detached documents from specific files. At each stage in the transformation, new files contribute to the content being viewed, while others' content disappears.

At the same time, we have also detached (individual, unique) authors from documents. In fact, the whole notion of a personal, individual author of a document is relatively recent, as [McLuhan, 1962] pointed out. (McLuhan predicted, 30 years before the Web, that electronic communication technology would eventually bring an end to individual authors and sequential text, and even foresaw a return to a more aphoristic style of writing).

ConTextual documents hold out the authoring possibility of allowing multiple contributors to be integrated into the appearance of a single, rendered text. Authors may now be asked to contribute chunks to a work, or to a ConText, rather than to
contribute an entire work. Also, in future, we hope to build the mechanisms such that
chunks need not be local to a given ConText, but may be aggregated, for instance,
from other servers (for more on this aggregative model of ConText’s possibilities, see
[schraefel, Plaice, Mancilla, 1999]

Limitations of HTML

In practice the effectiveness of poptext and related constructs is very dependent
on the connection speed and the speed with which the browser renders each
document. Current browsers were designed based on the implicit assumption
that the page jumped to will normally look nothing like the current page. The
current display is erased before the new one is rendered, and the window may
be empty for some time.

When we really are jumping to a totally different page, the empty window helps
the eye adjust to the new image. However, the whole point of this kind of
hypertext is that the new display is closely related to the old one. A delay of more
than a fraction of a second makes it much more difficult to perceive the
relationship between the old and the new pages.

Fortunately, with high-speed connections and the latest processors, the transition
is usually fast enough to give the desired animation effect. Eventually, browsers
could appear which take advantage of parallelism between the before and after
displays.

A more serious problem arises with longer texts, or texts with longer pop
sections. When we click on the arrow, we are taken (as described above) to a
page identical to the current one except that the indicated section has been
popped open. However, the browser, oblivious to any relationship between the
source and target pages, takes us to the top of the new page. If the section
popped open occurred in the middle of the document (which we had scrolled
down), it completely disappears during the transition.

The preferred behaviour would be for the text above the popsection to remain in
place, and the text below the popsection to move down and make room.
Unfortunately, there seems to be no platform/browser independent way to force
this behaviour with HTML or Javascript.

However, it is possible to implement a good approximation. We can alter the troff
macros so that the link generated by expanding the .SC macro has a name and
links to that segment. Then when we open a popsection in the middle of a
document, the opened section will begin at the top of the new page.

Conclusion
This paper illustrates the benefit of making documents versionable, even if only on dimensions controlling the Degree of Detail. What the paper also demonstrates is the need for authoring tools that make structuring either text chunks or legacy text into such renderable documents. If it takes as much effort to create intensional documents as it does to write them, few will be produced.

Indeed, one of the goals of intensional HTML has been to make an author based solution to rendering these intelligent texts. The macros for stretchtext, poptext and choptext are a start at these author's tools.

Acknowledgement

This was made possible by generous support form the Canadian Natural Sciences and Engineering Research Council (NSERC).

References

m. c. schraefel, "Talking with Antigone", PhD Dissertation (interdisciplinary), University of Victoria, Canada (1997).
m. schraefel, J. Plaice. B. Mancilla, "The Intensional Web", this collection.

Appendix: Examples

Examples of Poptext from original chunks can be viewed throughout schraefel's website at http://lucy.uvic.ca/shaka. A prototype of a single large document, deployed entirely on the ConText model is also available at http://lucy.uvic.ca/mc/proto/sampling

Examples of the troff/nroff implementation of poptext on legacy text can be found at http://lucy.uvic.ca/~wwadge/refman.html)