# PUTTING THE HYPER BACK IN HYPERTEXT

BILL WADGE
*Computer Science Department, University of Victoria*
*PO Box 3055, Victoria B.C., Canada, V8W 3P6*
*E-mail: wwadge@csr.uvic.ca*

M.C. SCHRAEFEL
*Computer Science Department, University of Toronto*
*10 King's College Rd., Toronto, ON, Canada, M5S 1A4*
*E-mail: mc@cs.toronto.edu*

We describe a set of front-end macros for IHTML which make it extremely easy to produce certain dimensions of hypertexts which Ted Nelson, in his original concept of hypertext referred to as "Stretch Text." However, by placing these macros within the ConText (intensional document) paradigm, we refine that concept beyond Nelson's vision as well, renaming the attribute in the intensional context "Poptext".

## 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 60's [2]. 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 simply as nonsequential writing, in which the reader "can go instantly in a choice of directions from any given point" [2].

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 specific

order. Catalogues and reference works are also written often for random access to a pre-existing target.

Still, individual articles, whether in a magazine or reference tome, are sequential in composition. These limitations are imposed by the physical properties of printed paper as a medium.

The Web, as medium, has not moved too far beyond its paper cousin when it comes to documents. It is still based on a very conservative, paper-like, 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. Unlike paper, these next pages can be jarringly dissimilar, a point to which we shall return. Still, at any given moment in a web document, readers have only a limited set of directions in which to proceed. They can scan the visible part of the current page; proceed linearly (scroll) through current page; or jump to one of the pages linked from the current one. This usually makes the previous page from which they clicked disappear from view.

In Nelson's words, referring to the current state of hyperlinks, "[these are] the minimal concession to hypertext that a sequence-and-hierarchy chauvinist could possibly make" [3]. It is hardly surprising, then, that some of 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. So far the main feature of the web is not so much what hyperlinks have done to create a rapid virtual thumb to flip through linked pages. Rather, it is simply availability: now the data from these paper-based magazines is online, and these vast amounts of material can be searched quickly. This is the benefit of the Web as engine, not interface.

**Intensional HTML**

The Intensional HTML (IHTML) project [4,5,7] 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 originally designed for software version management [1]).

At first sight IHTML addresses only one of the deficiencies of HTML (lack of versioning), which is not the most serious shortcoming. However, as schraefel showed in “A Thousand Papers for ISLIP ‘97” [4] multidimensional versioning makes possible the authoring of parametrizable documents that the reader can interactively reconfigure. This is something new: documents rendered by a user that can vary not only in terms of document content, but also in document structure. Each transaction with such a ConText (Conversational Text) becomes a choice made by the reader as to the direction of the next step to take in the process of browsing the document.

The result is that ConTexts can offer the reader a much richer choice of directions in which to travel, as well as the possibility of moving through smaller, more manageable texts, without losing the context of one's journey through the document space.

One of the components of Schraefel's ConText paradigm that makes this management possible is versioning document components on Degree of Detail, a more refined version of what Nelson refers to as "stretch text"

**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.

ConTexts envisions this notion of stretching text as exposing variable degrees of detail in a document, but not just by amount of detail of a given extensional text. These degrees of detail can also be versioned themselves across other dimensions such as Level of Expertise of any given document version; the stretched components as well can include additional components from other dimensions or other rendered structures depending on the user's chosen parameters. In other words, 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 from that reader's global, general and local, specific requests of the document.

We can formalize a Nelson-esque 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.

In 1998, we implemented this degree-of-detail extension of IHTML with the <iselect> construct. (a kind of case statement) to conditionally include portions of text.

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

> Computer screens will be the entire structure of tomorrow's publishing and libraries.

and we want the depth 1 text to read

> 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.

The following IHTML fragment

```
Computer screens will be
<iselect>
 <icase version=depth:1>
  not merely the basis of office systems and
  systems for writers
  but eventually
 </icase>
</iselect>
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:

```
Computer screens will be
<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). For generating content from scratch, schraefel has suggested a collaborative model of authorship, in which writers contribute document components, or "chunks" (e.g. aphorisms) which then are selected and collected in various ways according to the current document context.

The details of managing this multi-dimensional authorship model go beyond the scope of this paper. What we describe here instead is a definitive solution to the technical problem of dealing with legacy, extensional documents, and a provisional solution to some of the authoring problem for ConTexts (intensional documents from scratch).

The technical solution for legacy documents 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 a well-tested pre-existing system: troff macros.

Troff is an early typesetting program still supplied with UNIX. Troff has been largely superceded 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.

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
<iselect>
<icase version=depth:\\n+d>
..
.de DE
.nr d \n-d
</icase>
</iselect>
..
```

The legacy authoring methodology should now be obvious: take an existing document, in 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, this works very well (see Appendix A, note 2 for an example of such a parameterized document). One reason for the ease of applying these macros to well-written legacy is that the main points of the text are stated somewhere clearly and concisely, and text can be enlarged or contracted around these key points.

**Poptext**

We can use this wrapper approach to begin to implement the richer ConTextual Degree of Detail as well. We also here rename "Degree of Detail" 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 fit. It differs in two small but important 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 some additional text remains visible to give the reader a sense of whether or not they wish to pursue a particular point.

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 abstract 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, too: clicking on the "Stretchtext for the Millions" arrow may not bring (say) the macro definitions immediately into view; only the descriptive text may show up, and if the reader wants to see an implementation example, they can expand that component.

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).

The notion of expanding and contracting text alone is not a new idea; many word/outline processors allow the author to expand or collapse sections and subsections of a document. The same idea appears in, for instance, 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, in 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 intensional approach to documents as component collections, with the use of the types of macros descibed above, we can build a series of wrappers that can be applied to intensional document components, and thereby mix features like degree of detail and level of expertise with equal ease (for an example, please refer to Appendix A, no. 1). We leave how we will manage the visualization of these dimensional components to another time.

## Benefits

One of the goals of the ConText approach to digital documents is to allow the user to interact with the document space without leaving their frame of reference. Current HTML supports this in part with its Name/Anchor tag, allowing a link at one point in a document to refer to a point further down or up within (usually) the current page. In this respect, the user is still more obviously within the world from which they just clicked, rather than being jumped into another. Skillfully deployed frames have also helped users stay within the sense of a given space. Frames, however, are a much abused feature of HTML and have some navigational oddities for which many users dislike them, and many designers refuse to implement them. Thus, in many common HTML situations where one is presented with a single web

page in a single browser window, a link frequently takes a user to a new context where there may be only a tenuous relationship between the previous link and the current window. While we have not yet tested this formally, our early research supports findings well known in interface design: user comfort and confidence increases when the navigation space from one task to another remains similar. Standard HTML leaping often breaks this connexion [6].

In a single, non-framed web page ConTextual document using just Poptext, we can reduce the contextual breakdown that occurs in regular HTML. Where the document, for instance, is collapsed to all headers, the user can take in the scope of the document immediately. When a component is clicked, it is expanded within the current screen, rather than triggering a leap to another screen. The user does not lose the current context.

Our hypothesis at this stage is that, when the user can shrink or expand detail or shift level of expertise *within* the current document window, that is, can make these adjustments without shifting contexts, the opportunities for comfort, confidence and learning are increased. Poptexts makes even this single benefit of the ConText document paradigm available to the millions.

**Appendix A: Examples**

1. Examples of Poptext from original chunks can be viewed throughout schraefel's website at http://lucy.uvic.ca/shaka, as well as at http://lucy.uvic.ca/mc/proto/sampling

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

**References**


1. Plaice, J A and W W Wadge, "A New Approach to Version Control", *Software - Practice and Experience*, v19, no3, pp268-276 (March 1993).
2. Nelson, T *Dream Machines / Computer Lib*, Microsoft Press, Redomond, Washington, 1987.
3. Nelson, T. http://www.xanadu.com.au/ted/TN/WRITINGS/ TCOMPARADIGM/tedCompOneLiners.html
4. schraefel, m.c. *Talking with Antigone*, Ph.D. Dissertation, University of Victoria, 1997.

5. schraefel., m. c. ConText: Intensional document creation, delivery andretrieval. In *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria, Canada, pages 417-419, August 1997.
6. Shneiderman, Ben. *Designing the User Interface: Strategies for Human-Computer Interaction*, Third Edition. Addison-Wesley Longman, Reading MA. 1988.
7. Wadge, W. W. G. D. Brown, m. c. schraefel, and T. Yildirim. Intensional HTML. In E.V. Munson, C. Nicholas, and D. Wood, editors, *Principles of Digital Document Processing*, volume 1481 of Lecture Notes in Computer Science. Springer-Verlag, 1998.