



MULTI-HEAD MULTI-TAIL MOSAIC

Brian C. Ladd, Michael V. Capps, P. David Stotts, Rick Furuta



Abstract

*The explosive growth of the World Wide Web is attributable, in large part, to the simplicity of the distributed graph model it uses for information storage and retrieval. The model is simple, general, and ubiquitous; the reader may choose almost any browsing path. Information providers, though, then use every trick at their disposal in an attempt to increase their power over how readers browse their pages. Two major shortcomings of current Web protocols are lack of support for concurrent browsing and synchronization of multiple browsing paths. This paper presents a computer-aided instruction scenario where the author (instructor) must overcome these weaknesses. In traditional hypermedia systems, a link relates a single source node to a single destination node. Projects such as Xanadu [3] and Trellis [6] previously generalized links to allow connection of multiple source nodes to multiple destination nodes. This generalization, which we term a Multi-Head/Multi-Tail (MHMT) link, allows an author to create concurrent browsing paths in a document, and to synchronize those concurrent paths if desired. This has applications to groupware as well as to documents browsed by single users. We describe our solution embodied in MHTML as a straightforward extension to HTML for defining multi-head/multi-tail links and an experimental browser called MMM that implements the extensions. **Keywords:** Automata, hypertext, hypermedia, World Wide Web, Petri nets*

Introduction

The explosive growth of the World Wide Web is attributable, in large part, to the simplicity of the distributed graph model it uses for information storage and retrieval. The model is simple, general, and ubiquitous; the reader may choose almost any browsing path. Information providers are using every trick at their disposal to control the presentation of their data on browsers around the world. Instructional material with linearly linked document structures enforces a given reading path; browsing-path encoding in URLs permits tracking of a reader's path; and documents written as a single page ensure that readers see information in a specified order.

These solutions are ad hoc and limited; what is missing is a unified method for authors to express the browsing semantics they want over the information they are presenting. Such control has proven useful in other hypertext information systems (such as Trellis [6]) in such fields as soft-

ware engineering and computer-aided instruction.

The research presented here was prompted by two particular problems the current generation of Web protocols fails to handle: concurrent browsing streams and synchronization of browsing streams. Concurrent browsing streams are two or more paths of browsing, all of which are part of the same user's browsing session. Many current browsers permit the reader to follow a link in such a way that a new window is opened with the content at the end of the link; they do not, however, permit authors to include this behavior directly in their documents. Synchronization of multiple browsing streams is also important. Being able to advance only so far in one stream without advancing another or advancing multiple streams in lock step are examples of behaviors needing synchronization. Current Web browsers and protocols fail to address this (not surprising considering the lack of concurrent browsing ability).

Concurrency and synchronization allow authors to “program” the browsing path behaviors they want readers to follow, including:

- Multiple simultaneous pages
- Must-be-seen-before relationships
- Parallel browsing streams

The degenerate model of such multi-links is the standard single-in, single-out Web link type, which allows traditional unconstrained browsing semantics.

We have encapsulated this extended Web graph model in a novel browsing client, **Multi-head Multi-tail Mosaic (MMM)**. We have created **MHTML**, an extension of HTML with facilities for expressing multi-head/multi-tail links. The resulting graph semantics are equivalent in a formal context to those of a Parallel Finite Automaton (PFA) [5]. PFAs can be used in similar manner to Petri nets for specifying hyperdocument structure, as explored in the Trellis system [1, 7, 6]. Automata have been used to good effect in computer-aided instruction and other fields where control over the traversal of a graph is important. It is useful to note that neither the author nor the reader need be familiar with Petri nets to interface comfortably with MMM.

The remainder of this paper contains a CAI scenario which serves as the driving example, the design requirements for MMM, the language extensions for MHTML and the MMM browser, a comparison of HTML and MHTML solutions for our scenario, a look at our continuing and future research plans, and references.

Scenario

Authors of computer-aided instructional material have, traditionally, had great control over the order of presentation of their material. The power of hypermedia, though, is that readers can follow train-of-thought explorations in linked information structures. As CAI applications migrate to the Web, authors find HTML restricts

their ability to express orderings more complicated than a linear progression. Our methods allow authors to integrate some control of presentation order with this traditional browsing freedom. The generalizations of the link we have created allows a mixing of both capabilities.

As a concrete example, consider the following scenario: A pathology instructor is preparing a corpus of information to teach students how to read an autopsy report. For pedagogical reasons, the instructor has three goals: (1) Students should always be aware of the autopsy protocol (the summarizing cover sheet for the autopsy). (2) Students should refer to both the textual description and high-resolution photographs for most sections of the autopsy. (3) Due to the nature of the current case, students should not progress past the internal examination of the neck without having already seen the external evidence relating to the neck.

Looking at current Web protocols, there are two approaches to handling these goals: Adding information to the nodes (such as the autopsy protocol) so the node content is what the author wanted, or describing in the content of the nodes what the desired browsing behavior is and how to achieve it. For a more complete description of these approaches and comparison with the MHTML alternatives see the Comparison section.

Requirements

Current World Wide Web protocols use a general graph model consisting of nodes (pages) and links (embedded URLs) with all browsing semantics provided by the client browser. Links go from one page to one other page unless the reader uses a browser function to launch concurrent (unsynchronized) browsing streams. Note that it is possible, using a browser which can open new windows, for the reader to create the same browsing states as those specified with MMM; this would require the author describing the desired states to the reader (more discussion

of this can be found in the Comparison section below).

Providing authors with explicit browsing controls with the power of parallel finite automata requires (at a minimum) the addition of multi-headed links and multi-tailed links. Multi-headed links are links pointing at multiple nodes all of which are opened (in separate browsing windows) when the link is followed; multi-tailed links are links which are only active when the browser has all the input nodes open and which close all the input nodes when followed.

The MMM project's original goal was to test the usefulness of PFA semantics in the World Wide Web setting. As such, rather than spending our time building a complex client-server infrastructure, we chose to build a working prototype as soon as possible. To ensure an appropriate sample population, it was necessary to make that prototype acceptable to current Web users.

From a high level the Web architecture is very simple: a reader's browser requests a page of HTML marked-up text from the author's Web server and then renders it for the reader. Early on we recognized that in order for authors to be able to define the reader's browsing semantics at the time they compose the hypertext, it was necessary to modify either the author's server or the markup language they write in; it was also likely that some modifications would be necessary to browsers to handle more complex browsing patterns.

The decision to extend HTML rather than modify server code was made for three reasons: to keep control of browsing semantics as close as possible to the author; it was easier to code; and to encourage wider acceptance of our changes. Since one of the goals of this work is to allow authors to define the browsing semantics they want their documents to have, it seemed right to keep control of those semantics in the author's document rather than in some server configuration file. Since we recognized that browser modifications would be necessary in any case, it was

easier to limit our code changes to only one side of the client-server transaction if we could. Finally, modifying server code would mean that one version of one platform's server would be able to handle multi-head, multi-tail links. Unless we were willing to extend a large number of servers on a variety of platforms (and could somehow convince the world to switch to our new and improved servers), our changes would never be widespread. Another obstacle to widespread acceptance with server-side modifications would be the increased use of server resources to support multi-head, multi-tail links.

Mechanisms that fail to work across the range of browsers greatly limit the potential audience for Web pages that require those mechanisms. Fortunately for the designers of variant HTML syntaxes, the behavior of Web browsers is to silently ignore markup elements that are not understood. Thus the author who uses the Netscape <BLINK> environment can be assured that everyone will see the content of the passage, although its rendition may be static. It is worth emphasizing that the difference between a Web browser that shows blinking text and one that does not is presentational; the content shown is the same in both cases.

In adding semantics to the Web, we faced the similar problem of retaining accessibility to sites that did not choose to use the modified browsers supporting that semantics. As we considered the semantics of multi-headed, multi-tailed links, we realized that a sensible alternate semantics to concurrent and synchronized link access would be to permit sequential access to the links. We encode the new link specification as a new environment type (MHMT), place the parameters needed for presentation of the new environment into its beginning marker, and retain the normal markup specification for anchors (A). Thus a modified browser recognizes the added environment and understands the new semantics to be associated with its display. An unmodified browser ignores the environmental specifications and treats its contents as standard links.

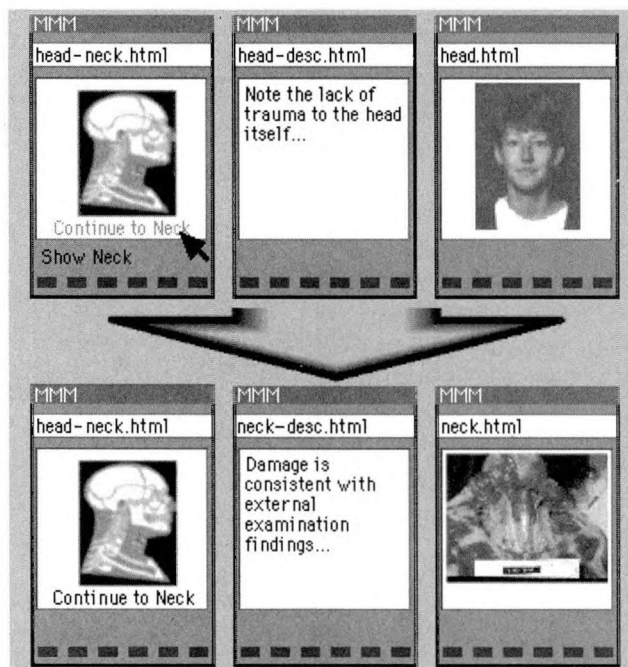


Figure 1: Following the "Continue to Neck" multi-link

MHTML: Extensions to HTML

Two considerations guided our extensions of HTML: Ease of use for authors and logic of presentation for unmodified browsers. Using existing constructs as much as possible forwarded both goals. The tendency of browsers to ignore unrecognized tags permitted us to accomplish the second goal fairly easily.

Example

A multi-headed/multi-tailed link is enclosed between a <MHMT> and </MHMT> pair of tags. Between these tags text, anchors, and our incoming reference (<IREF>) tag are all treated specially by our modified browser. An example link from the autopsy training scenario follows:

```
<MHMT text="Continue to Neck" display_
  text="Show Neck">
<A HREF="head-neck.html">X-Rays</A>,
<A HREF="neck.html">Neck Photos</A> and
<A HREF="neck-desc.html">Neck
  Description</A>.
```

```
<IREF HREF="http://www.cs.unc.edu/
  ~capps/mmm/head.html">
<IREF HREF="http://www.cs.unc.edu/
  ~capps/mmm/head-desc.html">
</MHMT>
```

This example has three incoming and three outgoing links. Incoming links are defined by the <IREF> tag, but remember that the file containing this MHTML definition is itself the head of one of the incoming links... so only two <IREF> definitions appear. Refer to Figure 1 to see how MMM displays and follows this multi-link.

The author's intention is that this link only be active if the current page (head-neck.html), the head photo page (head.html), and the head description page (head-desc.html) are all being displayed by the browser. Note that there are only two <IREF> tags; the current page is always implicitly in incoming link to any multi-link. As can be seen in the diagram, the multi-link is displayed in the browser window with the contents of the TEXT field in the MHMT tag. When the

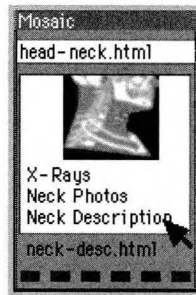


Figure 2: A multi-link in an unmodified browser

mouse pointer is over the active multi-link, the `DISPLAY_TEXT` field's contents are shown in the browser's status line.

The author's second intention is that the head photo and head description pages should be advanced in parallel to the neck photo and neck description pages. In terms of the MHTML, the current page and any pages referred to in `<IREF>`s in the multi-link will be closed and any pages listed as anchor references in the multi-link will be opened in separate windows. Hence the three anchors in the MHTML construct above lead to the three pages in the bottom portion of the diagram above when the mouse is clicked on the multi-link. Note that the multi-link is no longer active (display color went from green to black) since the two pages in the `<IREF>`s are no longer displayed.

So, with the MHTML construct the author has control of what the user sees, where the user is told that he is going (as opposed to the bare URL displayed by most browsers), the pages the user must have already visited (synchronization), and the pages the user will see afterwards (concurrent browsing).

As seen in Figure 2, an unmodified browser displays the three outgoing links as regular anchors. This is because they appear in standard HTML anchor constructs within the MHTML region and the MHTML region is ignored by unmodified browsers. The `<IREF>` tag is also ignored. Users of unmodified browsers are able to use an

MHTML document, though the author's concurrency and synchronization semantics are lost.

MHTML's Extended Syntax

```
Multi-link: <MHTML (tx | dtx | tx dtx |
              e)> body </MHTML>
tx: TEXT=text
dtx: DISPLAY_TEXT=text
text: any plain text string
body: (text | iref | anchor)*
iref: <IREF HREF=url>
anchor: <A HREF=url> text </A>
url: Universal Resource Locator
e: Empty string
```

The `tx` element is the text to display as the multi-link anchor text in the document and `dtx` is the display text to display in the status line when the cursor passes over the multi-link while it is active. If the `tx` element is omitted then the text within the MHTML block is rendered as normal HTML text; if it is present, then the text in the MHTML block is not rendered at all (somewhat like the description elements within an HTML 3.0 FIG block [4]). If the `dtx` is excluded the string "MHTML Anchor" is displayed in the status line when the anchor is active and passed over.

Prototype System

The Multi-Head, Multi-Tail Mosaic browser prototype is based on NCSA Mosaic 2.6 for X-Windows and compiles on multiple workstation platforms. Less than two thousand lines of code preprocess MHTML into HTML which the standard Mosaic renderer presents to the user. Hooks in the code

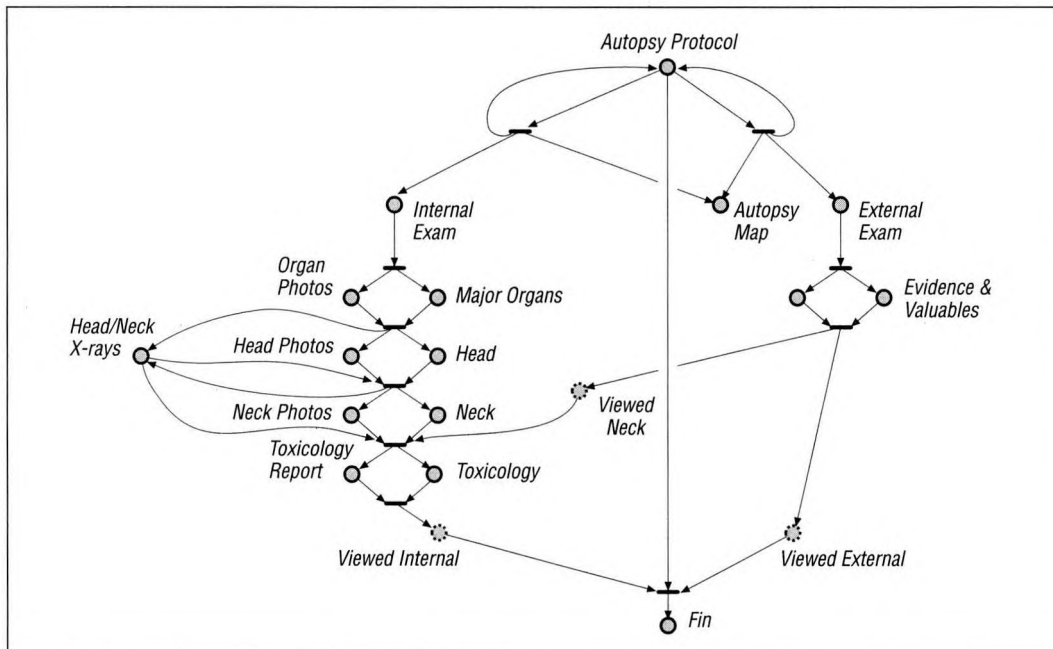


Figure 3: Overview of the autopsy report

present the DISPLAY_TEXT field when a pointing device passes over an MHMT anchor, open and close appropriate windows when one is selected, and update the status of MHMT anchors when windows open or close.

Screen dumps from MMM showing more details of concurrency and synchronization can be found at URL <http://www.cs.unc.edu/~stotts/elon/mmm.html>

The prototype MMM binary can be downloaded for experimentation at URL <http://www.cs.unc.edu/~stotts/MMM/index.html>

Handling the Scenario: HTML vs MHTML

Recall our example: A pathology instructor with three goals: (1) Students should always be aware of the autopsy protocol (the summarizing cover sheet for the autopsy). (2) Students should refer to both the textual description and high-resolu-

tion photographs for most sections of the autopsy. (3) Due to the nature of the current case, students should not progress past the internal examination of the neck without having already seen the external evidence relating to the neck. A PFA presenting the entire autopsy report as it should be read for study appears in Figure 3.

There are two approaches to structuring the presentation in HTML: change page content or describe how to create the next desired browsing configuration in the document. The first presentation goal of keeping the autopsy protocol visible for the student's entire session with the autopsy can reasonably be handled in either way. Since the autopsy protocol is relatively short it could be prepended in its entirety to every page in the corpus (a task potentially simplified by server-side includes). Alternatively, the author could place a statement above the links to the internal and external examinations that they should be opened in another window, perhaps including

instructions on how to do that with various browsers.

These approaches may well work in this case because the autopsy protocol is short (necessary for the first approach) and the desired browsing semantics are simple (necessary for the second approach). They would fail in the face of longer documents with arbitrarily complex browsing semantics. MHTML solves the first presentation goal with a minimum of hassle: each link to one of the examination pages is really a multi-tailed link specifying the autopsy protocol and the appropriate examination page as tail pages. The autopsy protocol does not participate in any other MHMT links so it remains up for the duration of the student's traversal of this material.

Either approach to HTML can be used with the second presentation goal that students see text and photos simultaneously for various sections. Modifying the appropriate sections to include the photographs inline takes advantage of one of the Web's most compelling features. In this case, however, the images are as tall as the Web browser's presentation area; scrolling up to see the picture and down to the descriptive text distracts the student. The instructor could include directives to the students to bring the images up in another window and to advance both the image and description pages as they move from section to section.

This second presentation goal is addressed directly by MHTML: links from section to section are multi-headed (including the description and image pages from the current section) and multi-tailed (including the description and image pages from the next section). For example, look at the link from Major Organs and the associated Organ Photos page to the Head, Head Photos and Head/Neck X-ray pages. (Note also that going from the Head and associated images to the Neck and associated images the Head/Neck X-ray page remains visible like the autopsy protocol above.)

The third presentation goal, synchronizing the browsing of two browsing paths, is much harder

to handle in HTML. It would be possible to modify the Neck section of the internal examination to include the text and photos of the External Evidence Photos and Evidence and Valuables sections, though if the student had seen them before this would be redundant. Alternately, a pointer to the Evidence and Valuables section could be included with an admonition to make sure it is seen before proceeding (it would then be necessary to handle a photograph and description as in presentation goal two).

In MHTML, browsing state is kept through the use of "empty" or "invisible" pages, which are marked but not presented to the user. Invisible pages are indicated in the PFA diagram by dotted outlines. Viewed Neck is one such page which is marked in parallel with the Belt Photos and Evidence and Valuables sections and not unmarked until it is used to pass beyond the Neck section of the internal exam. Hence if the student has seen the evidence information at some time during this session, it is possible to pass beyond the Neck section as soon as it is entered. If not, the student can follow the multi-link to the evidence description and photos, activating the link past the Neck section.

It is possible to get close to the desired browsing semantics using HTML, but the presentation is not quite what is desired (in the case of inlining images or prepending the overview) or it requires the author to describe how the reader's browsing should proceed. It is our belief that authors should concentrate on what to show, not how to get it shown. The MHMT construct in MHTML permits the author to describe the desired browsing semantics in a clear, compact method, concentrating on the interrelations of the data, not the user interface of the readers' browsers.

Conclusion and Future Work

We simplified the original design and implementation process as much as possible to allow us to test the feasibility and usefulness of multi-links in

Web browsing. We feel that we have proven the practicality of the MHTML construct and interface, and have seen numerous direct applications of its functionality, including the scenario discussed earlier in this paper.

While the original modifications to the Mosaic package were simple, they were made keeping extendability and flexibility in mind; indeed, the groundwork for some logical enhancements to our original design is already included. First and foremost, with the increasingly frequent new Mosaic for X releases, we found it necessary to make our changes easily portable to new versions. As new releases become supported by NCSA, we will update our patch files accordingly. Now that the feasibility and usefulness of MHTML have been shown we expect to extend the HTML parser to parse MHTML directly, removing the current preprocessor.

Development on increasing support for Petri net (and other graph type) attributes is continuing as demanded by trial users. Multiple marking (plural tokens) and multiple marking modes (colored tokens) are two examples of this. In addition, we hope to allow the user to select a specific graph model, like And-Or, Petri net, or General, for interpreting MHTML. Related to this, and certain to make MMM much more attractive to authors, would be creating a graphical authoring environment for building MMM documents or sites.

Probably the single greatest hindrance in reaching the goals of the project was the requirement that MHTML be handled correctly by an unmodified Web browser. We feel that one of our best directions for new development will be to remove that restriction.

Colored tokens, as mentioned above, are most commonly used in Petri nets for multi-user concurrent browsing. An example of this is running a moderated meeting using a deterministic order such as Robert's Rules [2]. Any sort of multi-user stateful model requires inter-user communication, either by modifying the Web server or causing the Web clients to communicate among themselves, at which point we have altered the client-

server model. Our original requirements called for no server modification, but we are investigating both of these methods. Future work includes implementation and experimentation with these new mechanisms. ■

References

1. R. Furuta and P. D. Stotts, "Interpreted Collaboration Protocols and their use in Groupware Prototyping," *Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94)*, Research Triangle Park, NC, October 1994, pp. 121-131.
2. R. Furuta and P. D. Stotts, *A Hypermedia Basis for the Specification, Documentation, Verification, and Prototyping of Concurrent Protocols*, Department of Computer Science Technical Report No. TAMU-HRL-94-003, Texas A&M University, College Station, Texas, 1994.
3. Theodor Holm Nelson, *Computer Lib/Dream Machines*, Redmond: Microsoft Press, 1987.
4. D. Raggett, *HyperText Markup Language Specification Version 3.0*, IETF Draft, 28 March 1995.
5. P. D. Stotts and W. Pugh, *Parallel Finite Automata for Modeling Concurrent Software Systems*, *Journal of Systems and Software (Elsevier Science)*, vol. 27, 1994, pp. 27-43.
6. P. D. Stotts and R. Furuta, *Petri Net Based Hypertext: Document Structure with Browsing Semantics*, *ACM Trans. on Information Systems*, vol. 7, no. 1, January 1989, pp. 3-29.
7. P. D. Stotts, R. Furuta, and J. C. Ruiz, *Hyperdocuments as Automata: Verification of Trace-based Browsing Properties by Model Checking*, *ACM Trans. on Information Systems*, to appear 1996.

About the Authors

Brian C. Ladd

[<http://www.cs.unc.edu/~ladd>]

University of North Carolina at Chapel Hill

Michael V. Capps

[<http://www.cs.unc.edu/~capps>]

University of North Carolina at Chapel Hill

P. David Stotts

[<http://www.cs.unc.edu/~stotts>]

University of North Carolina at Chapel Hill

Rick Furuta

Texas A&M University