Computer Networks and ISDN Systems 27 (1995) 823–829

COMPUTER
NETWORKS
and
ISDN SYSTEMS

Chapter 6, in which Pooh proposes improvements to Web authoring tools, having seen said tools for the Unix platform

Kate Weber [1]

Silicon Graphics, Inc., Mail stop 191, Mountain View, CA, 94043-1389, USA

Abstract

This paper proposes a series of improvements to Web authoring tools, in the hopes of stimulating the Web developer community to raise authoring tools to a higher standard. Web authoring tools are in their infancy. Most are little more than editors modified to output and display HTML. Few handle multimedia, integrate external conversion utilities, mimic the features of a server necessary to test the full function of the resulting HTML, or allow viewing the resulting HTML using more than one client. None handle more than one proposed extension to the current HTML standard. These and other features proposed here would improve authors' productivity, but more importantly the quality of their creations. When authoring tools integrate client and server functions, authors will be able to see their work as users see it, and this will allow them to better evaluate their material. In order to help the reader understand how these suggested improvements relate to existing capabilities, a review of authoring tools available on Unix platforms is included. The reviewed tools are HoTMetaL Pro, tkHTML, WWWeasel, WebMagic, and ASHE.

Keywords: Authoring; Editor; tkHTML; HoTMetaL Pro; WWWeasel; WebMagic; ASHE

1. Background and introduction

In May of '94 I took a job working on Silicon Graphics' Web server. I began to hunt around for tools to help my team produce and maintain good-looking, up to date on-line information. What surprised me was that there were so many individual conversion tools, and so few attempts at creating a single tool which would integrate existing capabilities (spelling checkers, version control) with Web-specific features (generating HTML, creating maps). Further, the capabilities of Web clients and servers themselves are not, for the most part, integrated into the tools I found. I began to think in earnest about improving the design of authoring tools, and decided to write this paper to propose the improvements I've collected to the Web development community.

I begin by giving examples of basic editing features, common in word processors and becoming more common in HTML editors. I then propose a definition of an authoring tool, which goes beyond the HTML editor model, and propose a set of authoring features. Following this I propose a set of advanced features designed to improve productivity. Finally I introduce those tools I've reviewed for this paper, list which of the features I've discussed each has, and explain in some detail what is implied by each suggested feature. I do not comment on how *well* these features are implemented, but merely focus on whether an attempt has been made. (I have

[1] katew@csd.sgi.com

0169-7552/95/$09.50 1995 Elsevier Science B.V.
SSDI 0169-7552(95)00041-0

not reviewed all the available tools, and especially I have not included a review of PC and Macintosh tools, because I believe the reader will form a clear idea of existing capabilities from the range I've reviewed.)

2. Basic editor features

Basic editor features are those which are commonly found in word processors, such as spelling checking and cutting and pasting, and the basic features needed to output HTML, such as applying tags to text blocks.

I set out in reviewing these tools with the assumption that all of the features I regard as basic editor features would be widely available. I did find each of the following features in at least two of the tools: cut and paste manipulation of text, highlighting a text object and applying an HTML tag to it, some method for making inserting links easier than just typing in the necessary codes, a help system, and templates. What surprised me was that the following features were not generally available: undo and redo, find and replace, a spelling checker, and allowing more than one file at a time to be open and displayed to the user. My intention here is not to focus on this level of features in these tools; I list them simply to better illustrate the difference between an editor and an authoring environment.

3. Authoring features

An authoring tool, to be really effective, should not merely be a fast and easy way to produce the desired output text, images and HTML encoding. It should allow the author to determine whether his/her design *works*, and to do so it must show the author how his/her work will be viewed and experienced by the reader. It must, in effect, "echo back" what is being implemented, so the author can see whether it is what they intended.

Viewing in client mode: The first component of the user's experience is the client used to view the material. The authoring tool should be able to display HTML to the author exactly as the client does. Further, since there are a multitude of clients and more arrive every month, the tool should have a configurable, expandable list of client views for the author to choose from. To me this suggests being able to launch clients for viewing side by side with an editor mode.

Editing: Whether an editing mode such as I just mentioned should be a raw ASCII HTML display, or an attempt at WYSIWYG, is an interesting question. It is much easier to position an image, for instance, by dragging it, than by typing in ⟨img align = top⟩. However, one could well want access to the raw HTML—for example, if the WYSIWYG mode doesn't support a particular HTML feature, or in order to correct an image or link reference. Existing editors take an interesting variety of approaches to this viewing and editing problem. The simplest tools allow editing in a raw HTML mode, and provide a simple preview mode which is not equivalent to any client. The more sophisticated tools allow editing in a "WYSIWYG" mode not equivalent to any client, and launch real clients for preview. One tool provides editing in a single client mode.

Editing in client mode: I recognize that this poses a significant technical challenge if client mode means one of any number of clients. A possible solution is to provide editing in a mode which nearly approximates a single client, and supports viewing in all other clients.

Multimedia: In addition to being able to preview and edit text, the author should be able to preview and edit images, sounds, video, etc. Since there are a wide variety of tools available for conversion and editing of various media, this implies allowing the authoring tool to be integrated with whatever tools the author specifies. Just as clients have the MIME types concept for viewing media not natively supported by their client, so too an authoring tool should allow the author to configure which conversion utility to use to import, for example, a word processed document, or which external media editing application to trigger to allow the author to edit media.

Allowing the author to follow links: Since Web pages don't exist in a vacuum for the user, an author should be able to select a link and follow it. The document root, an underlying part of the server software, makes this a challenge, since the reference in HTML to a file contains the path name of the file *from the document root*, rather than from the real machine root. The authoring tool must be able either to independently mimic this server feature (i.e. store the document root and resolve addresses), or query a Web server for the information it needs (mimic a client).

Resolving image references: Since images are not physically contained in the HTML files but are rather referred to by them, in order to display a complete Web page that involves images an authoring tool must resolve the image address and go get it. This is the same problem as with following links.

Graphical link representation and editing: Another way to assist the author in understanding the effectiveness of their Web information would be to represent the HTML pages as nodes in a graphical representation of the link structure. So, hierarchically arranged data could be represented in a tree, narrative information in a line, truly amorphous data in—what else—a web! The next logical step is to consider editing the information's structure in graphical mode. If the view of your information as a hierarchical tree allows you to identify the links you want to add, why not add them in that mode? One of the tools I reviewed does show a hierarchical view of links, but doesn't allow manipulation of those links.

4. Advanced productivity features

In addition to the above ideas, which I feel begin a richer definition of "authoring", I have the following ideas about how to make an authoring tool more effective. These ideas address making the tool more powerful, and thus giving the author more power in creation. These features include: syntax checking, supporting the creation of maps and forms, allowing for the use of more than one tag set, integrating conversion utilities, link maintenance, link checking, version control and security.

Syntax checking: While the authoring tool must, of course, put out valid HTML, it should be able to take in substandard HTML, notify the author of needed changes, and correct errors where possible.

More than one tag set: At any given time these days, there are server administrators worrying about when and if to start using the next HTML standard tags—when will the users adopt the clients that support these features? Just as many servers currently have one path for clients that support images and one for clients that don't, an author could want to generate, for example, both pages that use the Netscape extensions, and pages that don't. If the authoring tools approached the HTML tag set as the dynamic thing that it is, and allowed the user to designate which tag set to use for verifying the current page, the author could support multiple client environments.

Supporting maps: There are actually map editors available. For some reason, none have been integrated into any authoring tools. Support for maps could include generating coordinate numbers (with the cursor on a particular position on the graphic the user is informed of where on the x and y axis the cursor is), generating the map file automatically using coordinate positions chosen by the user, and updating the map configuration file. Again, integration of the authoring tool with the server function would allow the author to test his/her output.

Integrated conversion utilities: Let's say you are an author, and you want to import into your document an image you have stored in postscript format. You want to convert it to gif, and you happen to have a postscript to gif converter you like very much, although you've forgotten what it's called. Rather than making you remember the name, run the conversion outside your authoring tool and then do the importing, the authoring tool should allow you to configure what conversions to automatically run on various formats of data when you import them.

Version control / security: The main advantage I see version control offering Web authors is that it allows multiple authors to collaborate on a single server, without duplication or loss of work. Allowing different security levels for different parts of servers also contributes to collaboration by allowing "less authorized" authors to work on the same server as "fully authorized" authors.

Link maintenance: Various things can make it desirable to move a file or files from one location to another. When a file is moved, any HTML that refers to it must be updated. A really good authoring tool would look for, and update, references to files that are moved. (Naturally this only works on a server to which the tool has access. If someone in Austria has a link to my page in Japan, I probably can't help them, except by leaving a "forwarding" reference at the old address.) Note that if an authoring tool has both link maintenance and version control, it should be able to handle the case where a reference that requires updating is in a file currently being worked on by someone else.

Link checking: The author should be able to select a particular piece of HTML and have the authoring tool "follow" the links, retrieve the items referred to, and report any problems (page not found, reference is to an image but text is retrieved, etc.). A labor-intensive part of generating HTML is checking to make sure the link references are correct. In addition to wanting to follow particular links while writing HTML to verify the logical flow, the author needs to verify that no typos have been introduced, etc.

5. Review of the existing tools

Although I've reviewed several pieces of software here and this information could help you select which software would be most useful for you, I've not focused on many factors that you would naturally need to: what kind of support is available when you encounter problems; the user interface (e.g. clear structure in menus, menu choices well named, understandable icons, etc.); what kind of commitment to expanding the tool the author(s) has/have; whether it has bugs! In addition, some of the features I list as existing did not work in the versions I reviewed. The following tools are reviewed:

tkHTML was written by Liem Bahneman. It is a released product under the GNU Public License. I've reviewed version 2.1. It is described by the author as "a simple HTML editor based on the Tcl script language and the Tk toolkit for X11. It allows quick composition and editing of HTML-format documents, as well as a rapid way of converting existing text documents to the HTML format."

HoTMetaL Pro is a released product of SoftQuad Inc. There is also a free version, HoTMetaL, available via anonymous ftp. I've reviewed version 1.0 of HoTMetaL Pro for the Silicon Graphics platform.

WWWeasel, a product under development by Enterprise Integration Technologies (EIT), is quite exciting, although it currently does not have a planned release date. The version I reviewed did not have a number. EIT has an ambitious plan for their tool, with many more advanced features than the other tools I discuss.

WebMagic is a released product of Silicon Graphics. I reviewed the 0.98 beta release. From the marketing literature: "WebMagic Author, part of the WebFORCE(TM) software environment, is a powerful editor that makes creating media-rich World-Wide Web pages easy."

ASHE, A Simple Hypertext Editor, was written by John R. Punin and is Copyright property of Rensselaer Polytechnic Institute. I reviewed the beta release. The authors describe it thus: "ASHE was written using C language, Motif and NCSA HTML Widget. [... It's objective is] to serve as a tool allowing on-line creation of HTML documents with WYSIWYG features."

Tables 1–3 compare feature sets among current authoring tools; the following sections give a detailed explanation of what each feature is, provided as reference for the reader who wishes to determine which tool they might want to acquire and use.

Table 1
Editor features

TOOLS:	WWWeasel	tkHTML 2.1	HotMetaL Pro	WebMagic	ASHE
cut and paste text and tags	yes	yes	yes	yes	yes
undo/redo	no	no	yes	no	yes
find/replace	no	yes	yes	no	no
spell check	no	no	yes	no	no
multiple files	yes	no	no *	yes	yes
highlight text and apply tag	no *	yes	yes *	yes	yes
inserting links easy	yes	yes	no	yes	yes
templates	no	no	yes	yes	no
help	yes	yes	yes *	yes	yes

6. Explanation of editor features (see Table 1)

Cut and paste: Optimally, the user should be able to move and copy both text alone and text with accompanying tags.

Undo and redo: This is key for those of us who make mistakes!

Find and replace: Useful for deciding to use a different term throughout, etc.

Spelling checker: It's not surprising to see so many spelling errors and typos on the Web, given that this feature is not widely available.

Multiple files open at once: Especially valuable in a hypertext environment, where editing one logical document can mean editing forty physical files. * HoTMetaL Pro does allow more than one file to be open at once, but only one is displayed. This ruins the view of two physical files as one logical one.

The most basic HTML editor function is tagging: Can the author highlight a piece of text or a media object and apply an HTML tag to it? * WWWeasel uses a radically different metaphor to represent the HTML tags. It represents the tags in their hierarchical relationship to each other as vertical bars to the left of the affected text. Thus one does not highlight text and select from a menu to apply a tag. Rather one drops the cursor to the left of the desired text, and inserts the desired tag. The end marker, while there, is not displayed. * HoTMetaL Pro supports applying, deleting and modifying tags.

Links: How easy/intuitive/automatic is it to insert links? In addition to simply dropping the anchor tag on selected text, does the tool prompt the user for the desired link URL?

Does the tool include a template function? This would allow the user to define templates for various HTML pages that occur frequently. Obviously any editor would allow a user to open a similar file and save it with a new name—a template doesn't require the user to remember the names of similar files, and prevents accidental overwrite of the sample file.

Is there help available? This is more necessary for some applications I've seen than others! * HoTMetaL Pro's help depends on a client being installed, which the user must first configure. On the one hand, the user can select which client they like, on the other the user has to figure out how to configure help before they get any help!

7. Explanation of authoring features (see Table 2)

Raw ASCII HTML view: Does the tool allow the author to see the HTML codes used?

Table 2
Authoring features

TOOLS:	WWWeasel	tkHTML 2.1	HoTMetaL Pro	WebMagic	ASHE
raw ASCII HTML view	no	yes	no	no	yes
WYSIWYAG view	structured editor	yes (no images)	yes	yes	yes (no images)
WYSIWYAG edit	yes	no	yes	yes	no
"WYSIWYG"—at least one client view	yes	no	yes	no	no
editing in WYSIWYG mode	yes	no	no	no	no
many clients, configurable	no	no	is configurable, but only one at a time	no	no
forms support	yes (preview, some tags, no script)	none (no preview, tags or script)	yes (preview and tags, no script)	yes (preview, no tags or script)	yes (preview and tags, no script)
document root	yes	no	no *	no *	no
cgi script support	no	no	no	no	no
server-side includes	no	no	no	no	no
preview multimedia	yes	no	no	no	no
links—graphical view	yes	no	no	no	no
links—graphical editing	no	no	no	no	no

WYSIWYAG view: What you see is what you *almost* get. Short of viewing HTML as a client would, does the tool offer a view that hides the HTML codes and interprets them, formatting the text in some way?

WYSIWYAG edit: Does the tool allowing editing in the interpreted mode, or can the author only edit in raw ASCII mode?

WYSIWYG: Does the editor allow the author to view the HTML as it would be viewed by at least one client?

WYSIWYG editing: Does the tool allow editing in real client mode?

Does the tool allow the author to view the HTML as it would be viewed by a variety of clients? Is the list of client types of viewing configurable and expandable? (Can an author install a new type of client software and have the tool use it to view the HTML they generate?) I envision a configurable menu item here that has a pop-up list of all the clients the user has installed.

Does the tool support forms? Support for forms could range from correctly displaying/handling forms in each client's view mode, to a tags menu for the forms tag set, to generating a simple script to handle form input (for example a simple script to mail the contents of the filled out form to a specified e-mail address).

Does the tool allow the author to follow links between documents? This involves at the least allowing configuration of a "document root" variable. This would provide the enormous advantage of allowing the author to preview the HTML at hand in the context of the material it is linked to. * While HoTMetaL and WebMagic do not actually use a document root variable, they do assist the author in creating a document using a local URL then changing the URLs to the relative address on the real server. This allows the author to follow links, but adds another step in creating HTML, and doesn't address the problem in existing documents.

Does the tool support server-side includes? Server-side includes involve a code embedded in an HTML page calling a function from the server at delivery time. An example is the last update time of a file: when the file is requested, the server checks its update time and includes it in the HTML. If the authoring tool could either attempt to locate the server to test the included feature, or could actually mimic the server feature, the author would be able to verify the complete function of the page being created.

Does the tool support cgi-bin scripts? This support could range from attempting to execute referenced scripts (this again depends on being able to locate the server where the scripts are stored) for testing purposes to supporting the languages the scripts are written in. Again we have the MIME types concept: if the file extension is .pl, fire up a perl syntax checker.

Multimedia: Can the editor handle the display/previewing of images, video, and sound?

Graphical link representation: Does the tool provide a clear, graphical view of link structure, to assist the user in developing tree structures?

Graphical link editing: Having provided the user with a clear graphical view of their link structure, does the tool allow the user to "draw" a link

Table 3
Advanced productivity features

TOOLS:	WWWeasel	tkHTML 2.1	HoTMetaL Pro	WebMagic	ASHE
syntax checker	yes	no	yes *	yes	no
multiple HTML tag sets	no	no	no	no	no
entities	yes	yes	yes	no	no
map support	none	none	none	none	barely *
conversion	yes	no	no	yes *	no
integrate external conversion utilities	no	no	minimal	no	no
version control	yes	no	no	no	no
link maintenance/file moving	no	no	no	no	no
security	no	no	no	no	no
desktop integration	yes	drag text	drag text	yes	drag text
link checking	no	no	no	no	no

between two documents, then prompt the user to indicate where in the ''source'' document to link to the ''target'' document?

8. Explanation of advanced productivity features (see Table 3)

Syntax checker: Upon opening an existing HTML file, does the tool inform the user of any HTML syntax errors that exist? This feature allows the user to avoid creating substandard HTML, and to import and upgrade substandard HTML. However, it is important that the syntax checking not be so rigid that it defeats the purpose. * HoTMetaL can't always import a substandard file, except as ASCII text. In order to be most useful, the authoring tool should give informative messages and attempt to correct the problem if so instructed.

Can the tool handle more than one HTML tag set? This would involve allowing the user to configure which rule set to use in applying tags. This is especially valuable in these days of uncertainty about WHOSE extensions to HTML will become the new standard.

Does the tool handle insertion of HTML escape codes for composed characters?

Does the tool support creating maps? Support for maps would include: display of the map graphic, generating coordinate numbers (with the cursor on a particular position on the graphic the user is informed of where on the x and y axis the cursor is), generating the map file automatically using coordinate positions chosen by the user, and creating/updating the map configuration file(s). Note that correctly creating and updating map configuration files is dependent on the type of server the map is being generated for. See my later items on server support for more on this idea. *ASHE does support inserting the ISMAP code into the image reference, even though it cannot display the image.

Conversion: Can the tool convert text and media (image, video, sound) formats? * WebMagic is bundled with media conversion tools, which are not however part of the tool.

Integration of other display, edit, conversion and creation programs: Can the tool be configured to use applications the author specifies? Imagine here a definition similar to the client-side ''MIME types'' which the user could configure to specify which external conversion utility or media editing tool to spawn based on a files extension/type. * HoTMetaL Pro does have a list of two ''import'' filters to run when it encounters a file of substandard HTML, which implies to me they might be able to allow the user to specify other filters without too much more development.

Can the tool be integrated with revision control?

Link maintenance / file moving: If the user wishes to move a file in their Web server, can the tool locate all references to that file and update them? If the tool works with revision control, can it handle such an update when it encounters a locked file?

Security: Can the tool prompt the user for a password if a requested file requires a higher level of security?

Desktop integration: Can files of text, images, video, sound be dragged from the user's desktop into the tool?

Acknowledgements

Ideas for Web development are by their nature more group accomplishments than individual. My ideas for the future of Web authoring tools have probably been shaped by half the conversations I've had on the topic of the Web. However, I would in particular like to thank the authors of these tools for their contributions, and Jay Glicksman, Dave Ciemiewicz, and Fred Weber for their contributions to these ideas.