

WEB PAGE ANNOTATOR

Dale Reed* and Sirisha Garapati[†]

The increased presence of distance learning and on-line course offerings mean content is delivered using the Internet and viewed using web browsers. Students need to electronically make notes on what is presented and be able to share notes. Teachers need to be able to customize Internet-delivered curriculum as well as generate shared curriculum.

There are two phases to this project. The first, called *Web Notes*, gives the ability to associate a note with any web page, where successive visits to that page automatically cause that note to appear. It is an approach for annotating on-line curriculum, both for students and teachers, including presentation of notes similar to a threaded newsgroup.

The second phase is a content editor/filter and consists of free-form customization of a displayed web page, acting as an “overlay” that filters the original content. A user will select a portion of a page and substitute replacement content, modifying, deleting, or adding to what is already there. These changes will then automatically be applied when that page is next loaded.

A working prototype exists for Web Notes, while the content editor/filter is under development.

This work is sponsored in part by a NSF PFSMETE fellowship, DGE-9809497.

Need to Customize Web-based Curriculum - The Problem

Course content is increasingly being presented using a web browser as a delivery mechanism. Teachers using this type of curriculum do not have tools to customize it. Teachers' changes to curriculum must also be easily shared with colleagues who might be facing similar problems. Similarly students using web-based curriculum need to be able to “mark-up” their browsing experience, much as one would a text book or lab notebook. Students can also benefit from sharing their web-based experiences and observations (via shared Web Notes) in contexts such as cooperative learning groups or when constructivist pedagogy is being used.

Related Web Content Filters

At its inception the web was a mechanism for researchers to share findings and annotations of their work. A current web page annotating program called *Third voice* [<http://www.thirdvoice.com>] allows user notes to be shared, though information is stored on a proprietary server. The Interactive Paper Project [<http://lrsdb.ed.uiuc.edu:591/ipp>] also enables sharing comments on underlying, though it is forms based. A primary motivation behind web page content filters has been the elimination of banner ads, with a notable example being [<http://www.junkbusters.com>]. The site [<http://www.junkbusters.com/ht/en/links.html#blocking>] has a fascinating description of filtering history and software used to eliminate web-based advertising.

* University of Illinois at Chicago, EECS Dept. and Northwestern University, Learning Sciences.
reed@eecs.uic.edu, www.eecs.uic.edu/~reed

[†] University of Illinois at Chicago, EECS Dept., sgarapat@eecs.uic.edu

It includes links to customizable filtering programs such as WebFilter as well as links [http://www.junkbusters.com/ht/en/ijbfaq.html#companies] to some of the advertising industry's comments on this type of technology.

The Solution: Annotating Pages with Notes and Customized Views

Annotating web pages gives users the flexibility of associating information with a particular web page seamlessly *within* the browser environment. A non-digital analogue might be post-it notes applied to a page in a book. A post-it note could be copied and distributed, while the underlying page in the book remains unchanged.

Customizing the actual view of a web page takes this a step further. Rather than associating additional information that is ancillary to the page (a Web Note), the changes filter the view of the page itself. The non-digital analogue might be scribbling on a transparent sheet that can be used to overlay a page in a book. Again, the content of the book remains unchanged. The only change is to the user's view at the user's option.

Annotating Pages with Notes

Consider the example shown in *figure 1*, where a section of text is highlighted and a note is associated with it. Highlighting text gives the option of associating a note with that text. Choosing to create a note brings up the note edit window shown in *figure 1*. Notes can be organized by folder, and users can make them private (could be stored on users machine only), public, or group. The existence of public and group note attributes implies storage on a server so that this information can be shared. Created notes are then displayed in a threaded newsgroup-type presentation, where the display of groups of notes can be "exploded" or reduced through an outline format.

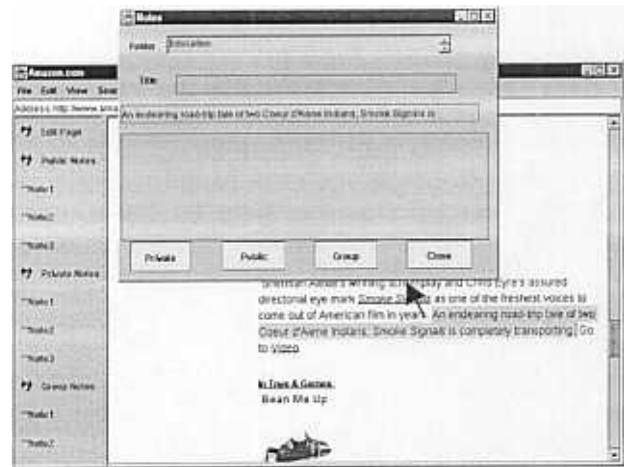


Figure 1: Web Notes Interface

Content Editor / Filter

The content editor window uses the same idea, though the relationship between the edit window and the browser window will be more complex. Consider the following example, for instance, where a teacher wants to add an item to a curriculum list given in some on-line curriculum. As shown in *figure 2*, the teacher first will highlight the section of the screen to be modified. The edit window contains the highlighted information, allowing modifications to be made. In this instance an additional item (Lab Notebook) is added to the list. Menus allow limited formatting of changes, and the changes will once again be given an attribute for private, public, or group sharing. The preview button will bring up a new window with a view of the changes, and once the "Submit" button is selected, the changes will be stored.

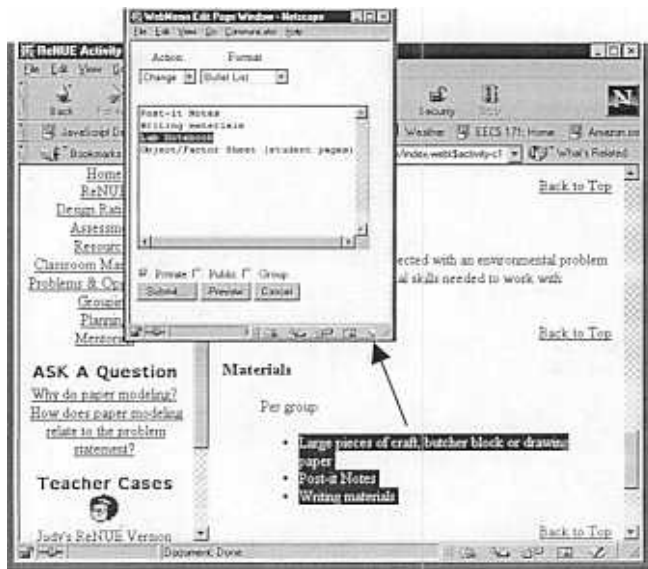


Figure 2: Selecting Content to Edit

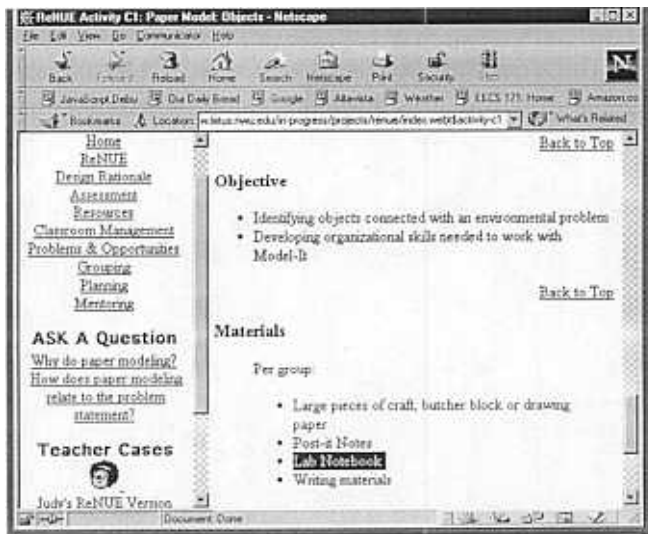


Figure 3: Filtered Contents Now Displayed

All web page names (URLs) pass through a filter to determine whether or not any changes have been stored for that page. When a page address is submitted to a browser where that page has been modified, the underlying changes are retrieved from a database. The database is either a flat text file on the user's machine (in the case of private notes) or stored on a server for group or public changes. The changes are then applied to the page before it is displayed, with the net effect that the page appears in the browser, changes and all, as shown in figure 3.

Future work: HTML Parser, Page Builder

The most difficult problem is to create a correspondence between a browser's graphical display and an underlying description of the HTML used to generate that display. Simple filtering such as that used in Web Notes can use a proxy server, intercepting the web page name and calling up an additional window when a note exists. A fully functioning content editor, however, implies the existence of an HTML parser to keep track of the underlying data structure associated with each displayed object in the browser window. We are in the process of modifying the Java HTML Editor class to give this. We would also like to create a page builder which could bring together pieces of curriculum from different sites, giving a personalized unified seamless view on otherwise distributed material.