

Using cascading style sheets to accommodate websites for individuals with low vision

Wayne E. Dick, PhD.

Dept. of Computer Engineering and Computer Science

California State University, Long Beach, CA

E-mail: wed@csulb.edu

Abstract

The Web Community tolerates bad markup for visual effect even when the practice excludes partially sighted users. This paper describes a two-phase process to recondition websites: deconstruction and building a new skin. It uses cascading style sheets. Deconstruction replaces developer styles with good defaults for partially sighted users. Building a new skin customizes the site for individual preferences. It adjusts font size, font family, color configuration, line height, word spacing and letter spacing. Its styles for lists and tables aid navigation. The process works best on sites that conform to W3C standards. Even marginal sites can be transformed so that the new interface is pleasant so long as the site developer does not misuse markup too aggressively.

Style bias on the Web

Web culture is visually dominant. While there are laws and strong moral reasons to address the needs of individuals with no vision or limited vision, the reality is that web culture tolerates improper use of web languages to achieve visual goals. This is true even if the misuse of web technology excludes many or all individuals with limited vision. Even well structured web sites plan little for blind or partially sighted readers. The non-visual subculture is simply unimportant.

This problem is exacerbated by a real confusion as to purpose of the web. Is it a medium for branding or conveyance of information? When a site considers branding its primary goal, then development skews toward graphical design. The development emphasis centers on making the site visually exciting and enticing. For a commercial site selling products, this might be appropriate. However, even corporate sites can suffer from burying services and sales transactions behind graphically attractive but functionally questionable interfaces. All too often, universities and public agencies feel the pressure to brand their agency first and relegate dissemination of information to a secondary status.

Given the normal challenges of reading with limited vision, partially sighted users can find sorting through marketing to find useful content overwhelming. The web that results from this visually dominant, brand driven culture is a platform that grossly under serves individuals with limited vision.

Creating an alternative view

This paper will describe how to recondition websites, so individuals with low vision can use them. This transformation use cascading style sheets exclusively. Depending on the degree of language misuse by the author, the style sheets will be more or less successful.

Pages that entangle too much presentation detail within the document structure will be impervious to the transformations. Sites that obey the latest HTML 4.01 or XHTML 1.0 strict standards or come close will become good user input for individuals with low vision. Even sites with presentations that are marginally usable, painful or even nauseating to individuals with low vision can be transformed so the interface is usable and even pleasant, so long as the site developers did not misuse markup language too aggressively.

There are some basic transformations needed by users with low vision. These include some combination of larger print, expanded spacing and control over font family and color [3]. Moreover, enlargement cannot be simple-minded zoom technology. It must wrap gracefully like a well-behaved word processor. Horizontal scrolling should only be necessary to move from column to column, not to navigate within one column. The page must use limited space wisely, and provide visible cues for traversal. Finally, and most important, the transformation for the individual with low vision must fit the users needs exactly. Thus, the user of a customized style sheet must be in total control of the style choices used to achieve their solution.

The style sheets presented here focus only on the video display, not print on paper. Print on paper is an inflexible medium that can trap partially sighted users into accepting a sub standard accommodation. This paper also does not consider aural or tactile modes of presentation. Both of these media formats have been the focus of considerably more research and development.

There are two phases of building an accommodating style sheet. The first is deconstruction. In this phase the original developer's style settings are overwritten with reasonable defaults for the user. The second phase is building the new skin, making the site pleasant and usable for the individual reader with low vision. These will be discussed in sections III and IV below.

Related development and research

While considerable effort has gone into developing aural interfaces and Braille access for the blind, relatively little software development has addressed the needs of individuals with low vision. Most assistive devices for this group focus on zoom technology, simple linear magnification.

To date only one product address the issue of intelligent enlargement effectively, WebAdapt2Me, released by IBM [3]. This product not only provides intelligent enlargement, good word wrapping with full webpage linearization as needed, it addresses the issue of adjustable spacing, color and font family. The internals of WebAdapt2Me do not rely on style sheets, but instead rely on manipulation of the markup using the Document Object Model. This gives the product much more control over output than a simple style sheet approach does, but it pays a price in performance.

Style sheets have been used by Silas Brown [1] to enlarge print in websites, and to rationalize the page presentation. These style sheets can be produced dynamically using a simple web based user interface provided by the author. The author provides little choice of color control. These pages do not offer spacing or font family as a choice, although this could be added easily. This site is an excellent resource for low vision users.

Clark [2] gives a good requirements analysis and guidelines for writing accommodating style sheets for individual sites. His paper is directed at developers. The World Wide Web Consortium points out those strategies that require extra coding by developers run the risk of being ignored [4].

The style sheets described in this paper target Internet Explorer level 6.0. They are only written for one user, the author. Nevertheless, they demonstrate some methodologies that can be generalized and are not present in other solutions. The concept of a formal deconstruction of the developer's original intent prior to applying a new skin appears to be new. Also, the use of new typography to meet the reading needs experienced by readers with limited space also appears to be new to the area of accessibility. Most authors of enlargement tools tend to enlarge HTML elements in a uniform way. This is not the case with the new skins presented here.

Deconstruction

Originally, this author wanted to preserve as much of the original page style as possible. After studying several websites it became clear that the style preferences of the visually dominant culture are just too incompatible with the style needs of the sub-culture with low vision. Even well structured pages developed for the dominant culture adhere to a different set of visual needs and expectations. Reluctantly and with some guilt this author decided that the style intent of visually oriented site developers needed to be nullified completely. This process is called deconstruction because it replaces the style conventions of the dominant culture with style needs of a subculture.

This process sets reasonable baselines for all HTML video display elements and their style properties. For example, it is necessary to disable all property settings tied to different "class" and "id" parameters for the same HTML element. While "class" and "id" attributes are used for creating the visually rich appearance that is required by graphic designers, they usually leave the page confusing and often unpleasant to the reader with low vision.

These are the basic guidelines for deconstruction.

- A few defaults may be set for all HTML elements. These include most font, text and spacing properties. So, the CSS 2 Standard wild card construct, “ * { ... } , ” can be used for these properties with no ill effect. Defaults can always be overridden for special elements by declaring them later in the style sheet.
- Elements that have text in their body, can be grouped together for default assignments. These include paragraph, table, list, body, frame and frameset elements. Grouping elements like `div` and `span`, also fall in this category. They all need word wrapping, readable font size, a clear font family, and a uniform alignment. The best font sizes for these elements are adjustable: “em” scale size.
- Block elements need to have the simplest possible visual layout. This usually requires the “`position: static`” setting and “`float: none`”. This renders linear pages as much as possible. Allowing objects to float right or left usually does not leave enough space for large print text. Most block elements should have their dimensions set automatically using the “`auto`” value for width

and height. Images and buttons are important exceptions. The author's sizes should be passed through.

- Anchors are an isolated case. Their usage in the general page typography tends to demand fixed font sizes.
- Forms and their input/output sub elements also require special care. They must stand out and function correctly. One bad side effect of enlargement is that it can expand vital controls right off the web page. As in the case of anchors, fixed font size defaults tend to work best. Forms that are bound within fixed width frames may be impossible to enlarge without loss of usability. (This author actually maintains many style sheets to accommodate pathological web sites. One special style sheet sets the font size for forms to the Windows default. The “`overflow: auto !important;`” command can help.)
- Headers do not have defaults. The styles of headers are style decisions for the new skin.
- Preformatted blocks need fixed font sizes as defaults. It is best to choose the smallest readable font size so that most lines can fit on a screen. For some word wrapping set the white-space property to “`pre-wrap`”.
- The information type elements like emphasis, bold, code etc. are part of the new skin.

The new Skin

General principles

The style sheets presented here have been customized to fit the visual needs of one person, the author. A full solution would require a user interface that enables customized selection of style by end users. Even this first attempt at custom style sheets for just one user was sufficient to reveal that one style configuration does not suffice for all situations. This author has between ten and twenty custom style sheets to meet all reading contingencies. So, a full solution to customized style for users with low vision also requires the ability to change styles dynamically to accommodate different reading environments.

Independence of presentation from structure in documents permits this flexibility. It allows almost limitless style manipulation as well as the ability to dynamically change the acculturation for each user and reading environment.

My Skin

Color, font, size and spacing

I live with low vision caused by prenatal damage to my central retinas. That means I need some strict conditions met on the basic parameters of style to read. I shift the basic screen color setting, from standard black text on white background, to black text on a tan background (red = 187, green = 153, blue = 102 in the CSS 2.1 RGB color settings). I strongly prefer sans serif fonts to serif. Tahoma is my favorite. I widen all possible

spacing: letter, word and line. The best enlargement of text for me lies in the range between 200% and 500% of 12 point font. I can live with 150% for short stretches, but when I'm tired 600% is best. Note that if any of these conditions are missing, color, font, spacing or size, I lose the ability to read long passages. One page is my limit before I start to hurt and feel sick. Warning: Just because this combination works for one person with central retina damage do not go out and declare this setting to be the optimum for individuals with low vision caused by central retina damage. I have friends with almost identical physiological damage who cannot abide with my setup. Style settings are very personal, and low vision marginalizes a reader's ability so much that any shift away from a personal optimum setting can render visual reading impractical.

Alternative adjustments enabled by separation of presentation from content

Markup language inserts markers directly into textural and graphical content to indicate the structure of a document. This parsing markup identifies the intended use of text and graphics within the document. For example, there are several levels of headers to denote the relative importance of sections. Also, basic content can be organized into paragraphs, lists or tables. Associated with markup files are style files. These files direct the rendering of markup elements for users. There can be many style files applied to the same markup file. This permits numerous intelligent renderings of the same content depending on the needs and preferences of the perceiver. This organization of document files is the basis of the World Wide Web Consortium's principle: separation of structure from presentation. It enables graceful transformations from an original medium and style to an unexpected medium and style. The same style sheets I use to rationalize the visual web to meet my personal needs can be used to make web pages present more rationally on a PDA. It permits a truly new skin to the same content.

Here are some features that work for this author:

- Headings are important semantic markers in a document. However, they convey far less information than the actual text. The visually oriented tradition makes headers larger than the text body because it makes them stand out visually and physically separates different sections of a document. This author appreciates the functionality, but when print is enlarged by a factor of four, larger headers are a waste of precious space. The solution is to make them smaller than the text. They still stand out from the rest of the page and provide physical breaks, but they waste no space.
- Lists are also problematic because they use indentation as a means of visual distinction. Multiple level lists employ multiple indentations. In the large print world, with wider than normal spacing indentation is costly in terms of space. My solution for this problem is to keep indentation but mark each list with a visible left border and force the width of each list item to a fixed size that is less than one screen width. This allows the reader to scroll horizontally to the left border of a list and then read the list easily with simple vertical scrolling to the end. If a nested sub list occurs then the reader only needs to align the sub list's left border with the left border of the screen to read the sub list. To go back to a parent list

from a sub list, just scroll horizontally left until the left border of the parent aligns with the left boundary of the screen. Then proceed reading the parent.

- Tables are probably the most difficult elements. I have found that table cells containing text can never be allowed to exceed the width of a screen. Otherwise horizontal scrolling within paragraphs is necessary, and reading becomes impractical. I find visible boundaries useful for traversing tables. Making the boundaries of tables distinct from cells also helps with navigation. This author can see color, so different colors help me. Other boundary differences work for readers who prefer grey scales.
- Boundaries can improve usability considerable. I create visible boundaries for paragraphs, headers and pre formatted blocks as well as table cells.
- If one can see it, color is a great tool. It permits visual distinction at a distance. I place my headers in a slightly different background than the rest of the text. Of all the style choices this is the most personal. So, use color with care. What is pleasant to one reader can nauseate another.

Peculiarities of Internet Explorer

All user agents have their quirks. Mozilla makes it difficult to change style sheets dynamically. Internet Explorer has more trouble than most user agents enlarging text effectively. Internet Explorer permits dynamic changing of style sheets and it allows the user style sheet to override the author's style settings without the use of the “!important” parameter for some properties. Here are some important things to note when developing style sheets for Internet Explorer.

- The max-width and max-height parameters do not work. So to control the width of table cells and list items one must place fixed numeric widths on “td” and “li” elements. Surprisingly this strategy works well on many pages. I use four widths so that I can fit one, two three or four table cells on a page at one time. Of course these four fixed settings require four separate style sheets. So, this is one way multiple style sheets come to be. A news paper front page is viewed best in four column mode. The text of a single article is best viewed in one column mode for reading.
- One can override the site developer's styles in Internet Explorer by using the [Tools>Internet Options>Accessibility] dialogue box. This box provides a set of toggle buttons that allow the user to override the site developer's settings for font family, font size and page color and to replace them to the Window's defaults. This dialogue box also allows the user to select a new style sheet from any style sheet available on the local computer. These two modes of overriding the original style result in some unique behaviors. Internet Explorer respects the CSS 2 guideline to give the site developer's style choices precedence over the user's styles with one big exception. If one chooses to ignore the developer's font family, font size or page color, then Internet Explorer uses the CSS 3 cascade to process the toggled style preference. The CSS 3 cascade order gives preference for user styles over the page developer's styles. Practically this means that if a user choose a toggle to ignore the developers style choice for a

property, then user must not use the “!important” parameter for the selected style property. All other style properties follow the CSS 2 rules and must use “!important” to override a developer’s style preference.

- The user’s style sheet may use “em” scale for font sizes whenever the page developer’s font sizes are ignored. This gives the user the nice option of using the built-in font-size selection option using the [view> font size] menu from Internet Explorer. Warning: Using “em” scale when the site developer’s settings for font size are still in effect can result in print that is too huge to be useful. The “em” scale sizes have an absolute maximum size. For larger font sizes, the user can accept the site developer’s font sizes and then override them to any fixed size using the "font-size" property with the “!important” parameter.

Conclusions

The World Wide Web is a visually dominant landscape, with a dominant culture that is profoundly insensitive to the needs on the sub-culture with limited vision. To mitigate this shortcoming of the majority development community, the World Wide Web Consortium has language standards that permit separation of document structure from document presentation. While the visual culture may be unaware of the needs of other sub cultures, this mechanism, if followed, permits rendering into numerous formats that can be customized right down to the personal needs of each user with special needs.

This paper has explored how this is done for one user, the author, on one user agent, Internet Explorer 6.0. While the example given here is of limited scope, it proves the concept that individualized accommodation can be brought to individuals with low vision. The process involves two phases: deconstruction and building a new skin.

There are several developments that could transform this proof of concept study into a functional solution for individuals with low vision. A good user interface is needed to collect user preference profiles. Linkage of this two step style sheet restructuring with a DOM translation program like WebAdapt2Me might improve both. The style sheets would perform better with well groomed source and the DOM analyzer could realize some cheap speedups. Finally, a professional user interface could be developed for optometrists and other treatment specialists who serve individuals with low vision. Careful selection of web style preferences could then be one additional part of fitting individuals with assistive technology for low vision.

Bibliography

1. Brown, Silas, Style sheets for low vision, <http://www.cus.cam.ac.uk/~ssb22/source/lv-css.html>
2. Clark, Joe, Big Stark and Chunky, CSS articles and tutorials A List Apart, ISSN: 1534-0295. 11 January 2005 – Issue No.
3. Hanson, Vicki L., Richards, John, The User Experience: Designs and Adaptations, ASSETS'04, October 18–20, 2004, Atlanta, Georgia, USA. Copyright 2004 ACM 1-58113-911-X/04/0010
4. Henry, Shawn Lawton (editor), Essential Components of Web Accessibility, <http://www.w3.org/WAI/intro/components.php>