A Web Accessibility Service: Update and Findings

Vicki L. Hanson
IBM T. J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY 10532 USA
+1 914-784-6603
vlh@watson.ibm.com

John T. Richards
IBM T. J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY 10532 USA
+1 914-784-7731
jtr@watson.ibm.com

ABSTRACT
We report here on our progress on a project first described at the ASSETS 2002 conference. At that time, we had developed a prototype system in which a proxy server intermediary was used to adapt Web pages to meet the needs of older adults. Since that report, we field tested the prototype and learned of problems with the proxy approach. We report on the lessons learned from that work and on our new approach towards meeting the Web needs of older adults and users with disabilities. This new software makes adaptations on the client machine, with greater accuracy and speed than was possible with the proxy server approach. It transforms Web pages "on the fly", without requiring that all Web content be re-written. The new software has been in use for a year and we report here on our findings from the usage. We discuss this approach in the context of Web accessibility standards and Web usability.

Categories and Subject Descriptors
H.5.2 **Information Interfaces and Presentation**: User Interfaces – *Input devices and strategies.* K.4.2 **Computers and Society**: Social Issues – *Assistive technologies for persons with disabilities.* K.4.2 **Computers and Society**: Social Issues – *Handicapped persons / special needs.* K.5.2 **Legal Aspects of Computing:** Governmental Issues – *Regulation.*

General Terms
Design, Human Factors, Standardization, Legal Aspects.

Keywords
Web Accessibility, User Interface, Standards.

1. INTRODUCTION
At the ASSETS 2002 conference we reported on work we were beginning that had the goal of becoming a Web Accessibility Service [5]. Our design at the time was guided by a number of requirements from interviews with focus groups. The actual software was, as yet, untested. In the two years since that conference, the service we envisioned has undergone considerable change. We report here on lessons learned and on the current state of the research effort.

Our work was initially motivated by a desire to make the Web more accessible for older adults. As we age, numerous sensory, physical, and cognitive changes occur. These changes, in many cases, can limit our ability to utilize the Web [4], [11]. Decreases in acuity, contrast discriminations, and color perception all impact ability to see the content on Web pages. Arthritis and tremors that are common among older adults impact ability to use a mouse and keyboard. Additionally, memory impairments combined with lack of domain experience make it difficult to understand Web navigation.

We sought to create a system that could rapidly transform any Web page to make it more usable for people with visual and dexterity limitations. The ever-increasing complexity of Web pages has resulted in ever-decreasing accessibility. Moving from its origins as simple hypertext markup, the Web has become populated with pages having eye-catching designs and rich, multi-modal, and dynamically generated content. Complex interaction paradigms have emerged as Web usage has moved from simple browsing to searching, shopping, and communicating. The technical complexity underlying this explosive change in the Web has served to undermine its accessibility and its usability. Our goal was a Web Accessibility Service that would adapt to such dramatic changes.

We were motivated, further, by the observation that assistive technologies, while indispensable for users with disabilities, also have many downsides. Chief among these downsides is the fact that they generally are designed to address a disability in isolation. As a result, they are not tested with other assistive technologies and users often find themselves with technologies that are incompatible. Unfortunately, disabilities commonly co-occur in individuals, leading them to need multiple pieces of assistive technology – pieces that may not have been tested together. Our goal was a Web Accessibility Service that would deal with multiple, varied, and fluctuating user limitations in a single service.

At ASSETS 2002 we described a prototype service about to be tested in the field. The defining feature of that service was that the adaptations to Web pages were made by an intermediary proxy. By centralizing all content transformations, and keeping software off individual client machines, we hoped to be able to rapidly adapt the service as Web content continued to evolve. In the next section, we discuss issues that arose as we tested this prototype in the field.

ASSETS'04, October 18–20, 2004, Atlanta, Georgia, USA.

2. PROXY SERVER AS INTERMEDIARY

In our prototype, Web content was transformed by having all HTTP requests flow from individual browsers through a proxy. The pages returned by the server to the proxy were parsed and transformed on the proxy according to users' specifications and returned to the requesting browser. The advantage of this architecture was that it required no modifications to the client machine other than the setting of an HTTP proxy in the browser. In principle, all browsers on all platforms should work equally well using this approach.

We implemented a prototype of a transcoding proxy server per the architecture described in our ASSETS 2002 paper [5]. This prototype was tested for several months by members of the SeniorNet organization [16]. SeniorNet provides not only an online community for older adults, but also supports computer centers where older adults can take classes. For the testing of our prototype, we worked with older adults who were instructors and students in Internet classes at SeniorNet Learning Centers.

Despite the promise of this approach, the testing uncovered significant problems using a proxy server. While users were generally positive about the project goals, their use of the system was severely limited by problems inherent in the architecture.

2.1 Correctness of Transformations

The greatest problem we faced was accurately transforming Web content. In order to do this correctly, our proxy had to become as capable as a full-function browser, correctly interpreting HTML, JavaScript, cascading style sheets, and a large and growing number of third-party plug-ins. This posed a considerable challenge.

Moreover, as we tested the prototype we learned that the majority of Web content does not fully conform to published standards. Browsers are very forgiving of most errors in the source documents, interpreting malformed HTML in some sensible, but idiosyncratic, way. The rules governing this interpretation are embedded in the browser's program logic, however, and are visible only by their effects. The need to accurately transform some of the content, while preserving the remainder in a way that allowed the browser to behave as it otherwise would, further complicated our task.

Given the difficulty of performing all the functions of a full-fledged browser, it turned out that a large number of Web pages processed by our proxy server were not correctly rendered on the users' machines. To increase the accuracy of rendering would have been an extremely large effort. We found ourselves trying to correct errors on a page by page basis, which clearly was not viable given the sheer number and continual updating of Web pages.

Other attempts to circumvent such ambiguities when using this intermediary transcoding approach have required that Web sites be annotated for presentation [2], [17]. Most Web sites, however, particularly legacy sites, will not be annotated, so this approach cannot meet our project goals of making the whole Web directly accessible.

2.2 Secure Web sites

A second problem for the transcoding proxy was created by secure sites. A proxy cannot see and modify the content flowing through a Secure Sockets Layer (SSL) connection unless that connection is mediated by a pair of SSL sockets at the proxy itself. Through this mechanism it is possible to see and modify the content on the proxy. Performing such an operation, however, violates the end to end security expected of secure connections, and obligates the proxy to guarantee the safe-keeping of the unencrypted content (technically difficult no matter how briefly the content is "in the clear"). Such an action also generates messages to users about the security violation which, understandably, raises concerns.

2.3 Setting the proxy

There were also problems with potential users trying to connect to the proxy. In order to set a browser to go through a proxy, it must be possible to set the HTTP proxy on the browser. This is not always possible since proxies cannot be cascaded. Thus, users who had a proxy already set, such as for some corporate firewalls and some ISPs (e.g., AOL), were prevented from using our transcoding proxy.

2.4 Copyright

The proxy server also created questions in relation to copyright since the server was making changes to Web pages and distributing these changed pages. As one way to address this issue, we adhered to the 'no-transform directive' [9]. This directive states that pages which include the header may not have their content modified by the proxy. During the period of our prototype testing, we found few pages that contained this directive. During a three month period of usage, we logged the URLs requested by users (but not who requested any given URL). Of the nearly 54,000 unique URLs requested by our users, less than .01% contained this directive.

2.5 Problems of scale

The prototype testing further indicated that a production version of the system would have required tremendous server capacity. Even with a powerful server and a relatively light load, users of our prototype complained of delays in page presentation. People familiar with the Internet are accustomed to a relatively high speed of page presentation. The extra time introduced by the additional socket connection from the proxy to the server and the delays associated with the transcoding on the proxy produced an unacceptable delay. The type of server capacity that would have been required for a viable production system would have been expensive, indeed.

3. A REVISED ARCHITECTURE

When taken together, it became clear that the proxy approach for a Web Accessibility Service was not viable. This caused us to reconsider our architecture. While moving the transcoding away from the client had some theoretically desirable features, these benefits had to be rethought in light of technical feasibility. We therefore moved to an architecture in which the transformations were performed on the client.

There were two critical design points for the revised architecture. The first was to let the browser do as much of the work as possible. Thus, we could sidestep the problems of trying to correctly interpret the vast variety of complex and malformed Web pages, of dealing with encrypted and copyrighted content, and of providing adequate server capacity. The second, and related point, was to avoid source transformations altogether, relying instead on the browser's own internal representation of the Web content.

As shown in Figure 1, our current approach involves a client and server architecture that transforms the content of HTTP requests without a proxy server. These transformations are performed on the client machine using a combination of approaches involving Document Object Model (DOM) manipulations and utilization of browser and operating system features. No changes are made to the page source. The server is used merely for storing user preferences.

Based on user input about the difficulties they experienced with the Web, we developed several different content transformations. One set of transformations was designed to address the visual presentation of pages. Certain changes such as font enlargement, font style, increased inter-letter spacing and inter-line spacing, and enhanced color contrast can increase legibility for many people [3], [10]. For users with greater degrees of visual problems, we provided additional changes, augmenting the text with speech output and providing options for very large banner-text displays, enlargement of browser controls, and whole-page magnification.

Adaptations were also developed to target limitations in hand usage. Some Web activities, such as writing e-mail, filling in forms, or completing login and registration information, require the use of a keyboard. Appropriate keyboard accessibility settings can significantly improve the accuracy of typing for some users, yet these settings are difficult to learn and manipulate [19]. To address this problem, we provided users with easy ways to configure their keyboard settings.

3.1 The DOM and Browser Helper Object

Certain dynamic visual changes to Web pages are accomplished via manipulation of the DOM produced by the browser itself. The DOM provides a model of an HTML document and allows the model to be changed in order to alter the page presentation. Our Internet Explorer® implementation uses a Browser Helper Object (BHO) written in Java, which gives program access to the document object before it is rendered in the browser (but after any JavaScript or other dynamic changes to the document have been made). Existing DOM APIs support both document manipulation and handling of user interface events in the document via the BHO. We manipulate page elements after the DOM is constructed. In this way, we overcome the problems of our previous architecture that produced a number of inaccurate renderings of Web pages.

The following visual transformations involve DOM manipulations:

- changing colors (text, background, and links)
- speak text
- largest text sizes
- very large 'banner text'
- image enlargement and enhancement
- page layout (linearization)

3.2 Browser and Operating System Features

In some parts of our implementation we exploit features built into the browser (e.g., certain font size manipulations and under-the-covers use of style sheets) and operating system (e.g., StickyKeys, FilterKeys, and the size of browser controls).

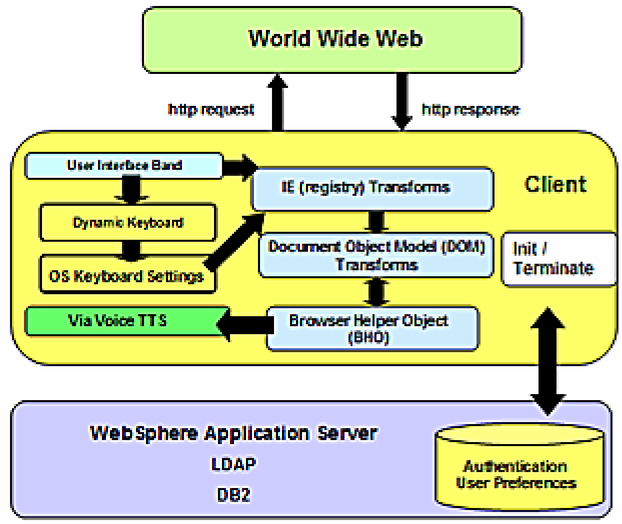

Figure 1. Revised architecture with transformations and adaptations performed on the client.

Sometimes this involves a straightforward setting of these features by users, as in the case of font style. For font style, there are choice buttons on the interface showing different style options. Users simply click a button to indicate the font style they wish to have for page presentation.

User requirements for some typing options are inferred from short samples of a person's typing. Thus, as a user types, our program makes a determination of optimal keyboard settings referencing typing models tested with users having a variety of keyboarding difficulties [18]. Based on this analysis, certain keyboard accessibility parameters are automatically adjusted to improve typing accuracy for the individual.

The following visual transformations and keyboard adaptations exploit browser or operating systems features:

- magnify pages (using style sheets)
- some text size increases
- text style
- line spacing (using style sheets)
- letter spacing (using style sheets)
- hide images
- stop GIF animations
- remove background images
- large browser options (pointers, controls)
- keyboard (StickyKeys, key clicks, MouseKeys, plus typing adjustments)

3.3 Setting and Saving User Preferences

As shown in Figure 1, user preferences are stored in a database on a server. This particular design point arose from requirements of the users participating in our research. Many of these people used the Internet in classroom or other shared-computer environments. Storing user preferences on a server allows multiple users to share one machine without having the preferences from one person impact the session of another. Conversely, it allows a person to

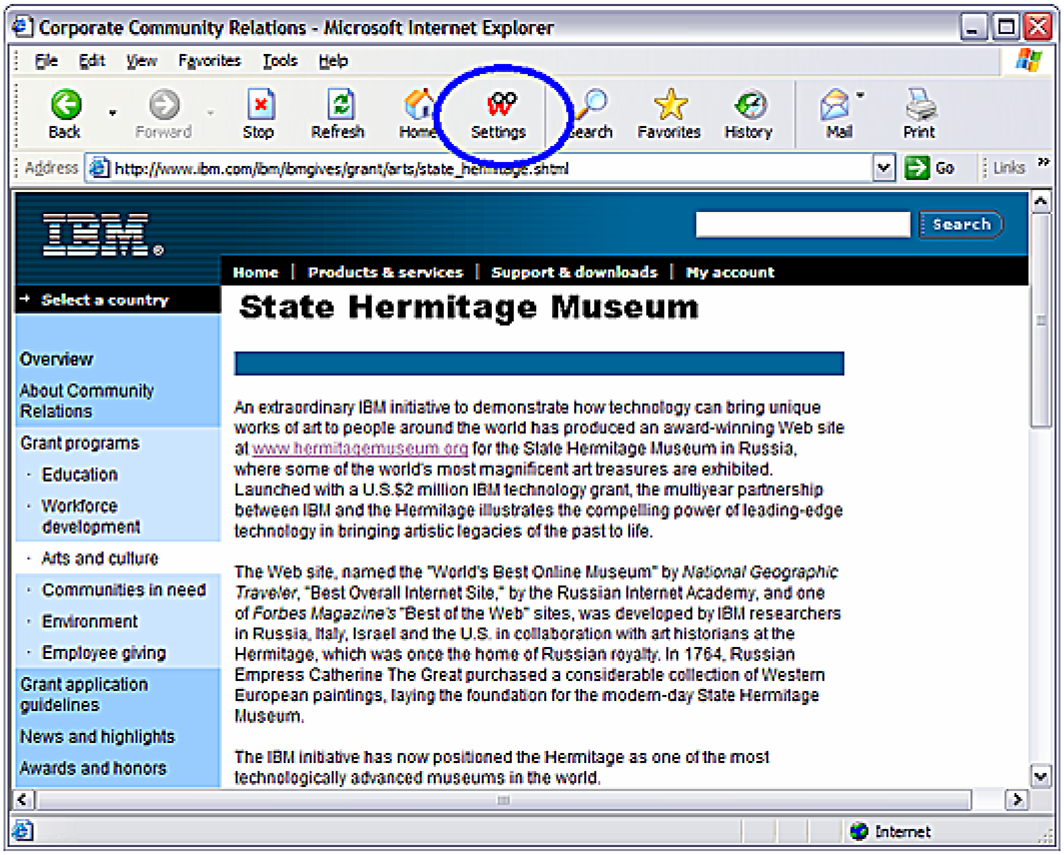

Figure 2. A 'Settings' button is added to the Internet Explorer® toolbar. Clicking this button brings up the settings panel.

have the same settings consistently applied when they use different machines (either, for example, different machines in a classroom or machines in a classroom and at home).

To begin an Internet session using our software, the user first logs in to a server. The logon screen is designed to be usable by persons with limited vision and keyboarding ability. Specifically, the screen completely fills a 640 x 480 display, with extremely large text indicating where people should enter their username and password. The typing adaptation software described above is active and with little typing input is capable of making adjustments to help users more accurately enter their information.

Upon successfully logging in, returning users have their settings preferences retrieved and applied. If they are still happy with those preferences, they use the Web as usual. At any point in time, they can change any of their preferences. New users specify their initial preferences. At the close of each session, the user's current settings preferences are sent back to the server and saved for the next session.

4. USER INTERFACE

Our user interface design is based on several specific requirements.

4.1 Standard browser

First, people tend to prefer a standard browser with the accessibility adaptations added rather than a specialized browser. As an early prototype, we built a simplified browser that used the Internet Explorer® engine, but had a unique limited set of icons on a toolbar designed to be easily visible and easily used by novices. This specialized browser proved to be unpopular. For instructional purposes, it was unpopular since it did not match existing materials for Web browsing. From a user perspective, it was unacceptable to users who had prior experience with the Web. They complained that this 'dumbed-down' interface limited their browsing experience compared with a standard browser.

4.2 Flexibility

The term 'dynamic diversity' has been coined to describe the abilities of older adults [6]. This phrase nicely encapsulates the dual aspects of designing for this population. They not only are diverse in their needs, but their needs can fluctuate from day to day and even within a session due to fatigue or a variety of other factors. Many will have a complex combination of needs. These users, therefore, require rather substantial flexibility in selecting and combining adaptations. As a result, any system must build in a great deal of flexibility.

4.3 Interactive and simple

Many of our initial users were new to computing and very few were skilled computer users. Consequently, it was important that the interface be extremely simple to set preferences. This user requirement combined with that fact that most people are

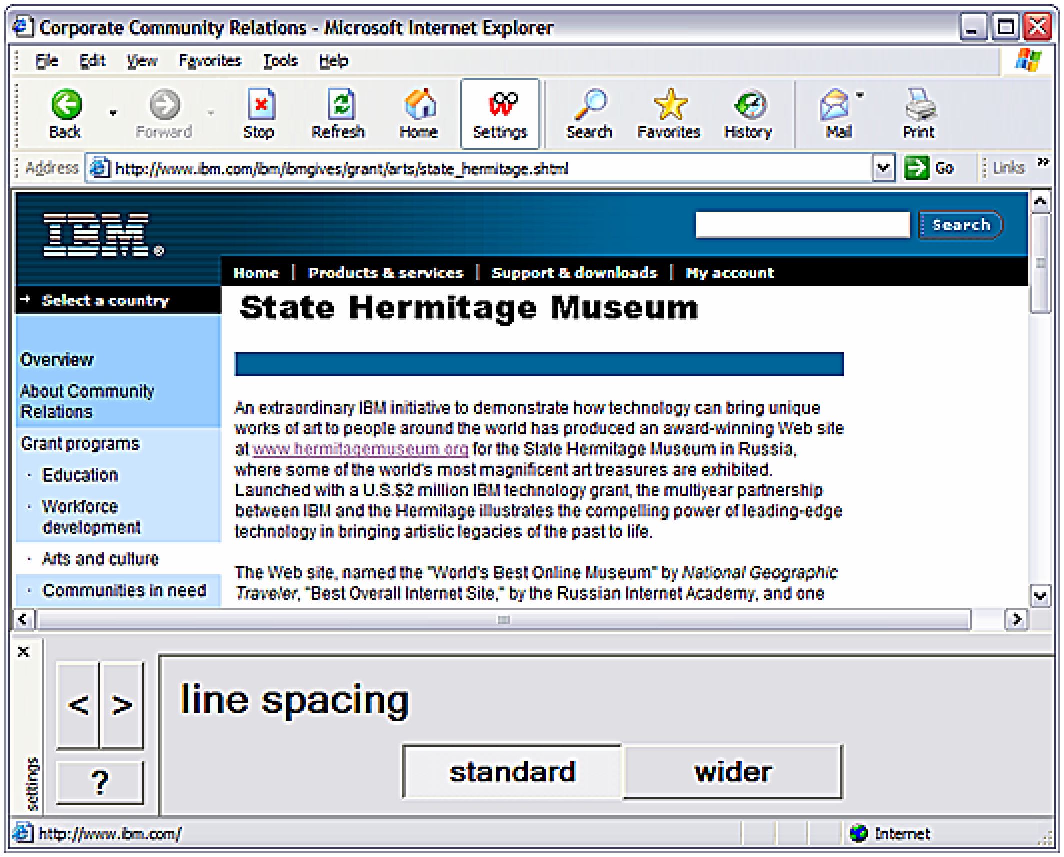

Figure 3. The Settings panel for 'line spacing'.

generally unable to 'picture' ahead of time what effect an adaptation might have on a Web page. Our solution therefore needed to make it easy to directly *try out* any adaptation on any Web page at any time.

4.4 The settings band

To address these user interface requirements, we designed an interface in which users cycle through a simple series of panels in a control band, with each panel focusing on a single type of transformation.

As shown in Figure 2, people use the standard Internet Explorer® browser. The single interface change is that a 'Settings' button is added to the toolbar. To bring up the settings band, a user clicks the 'Settings' button. The control band appears at the bottom of the browser, as shown in Figure 3. This Figure shows the panel in the control band for 'line spacing'. Let's use that as an example of how the panels work. To create more blank space between the lines of text, the user simply clicks on the button labeled 'wider'. As soon as the 'wider' button is clicked, the requested change is applied to the current Web page. The result is shown in Figure 4. If the user does not like the results of the change, they can easily go back to the way it was by clicking the 'standard' button.

Access to all transformations (except the typing adjustments) is provided through a different panel. Users can cycle through the panels using the arrow buttons on the left of the band. Interactive help, keyed to the particular panel, is available by clicking on the '?' button. When the user is satisfied with a particular adaptation or a set of adaptations, they can close the band (by clicking the 'Settings' button again) and browse the Web as usual. All subsequent Web pages that are viewed will have the selected adaptations automatically applied. Transformations are applied whether the band is present or not.

The ease with which users can test different transformations means that they can see which work best for them. This is particularly important in cases where users may not know in advance which adaptations will prove beneficial. In fact, users may discover useful transformations that they may not have predicted in advance of trying them out. Consider the line spacing used as an example here. During our focus group interviews, no one requested this feature. The literature, however, suggested that this adjustment would increase legibility [3]. When given the option to try out this transformation, many users did indeed discover that it made their Web browsing easier. A design that assumed that users could pick out their needed transformations from a checklist, for example, would likely not lead to this outcome.

5. USING THE ADAPTATIONS

The client version of our software has been in use for just over a year. Although the initial intent was to develop a tool for older adults and all preliminary testing of the proxy version was with older adults, we now find that the software is useful for a much more diverse user group. A number of organizations serving

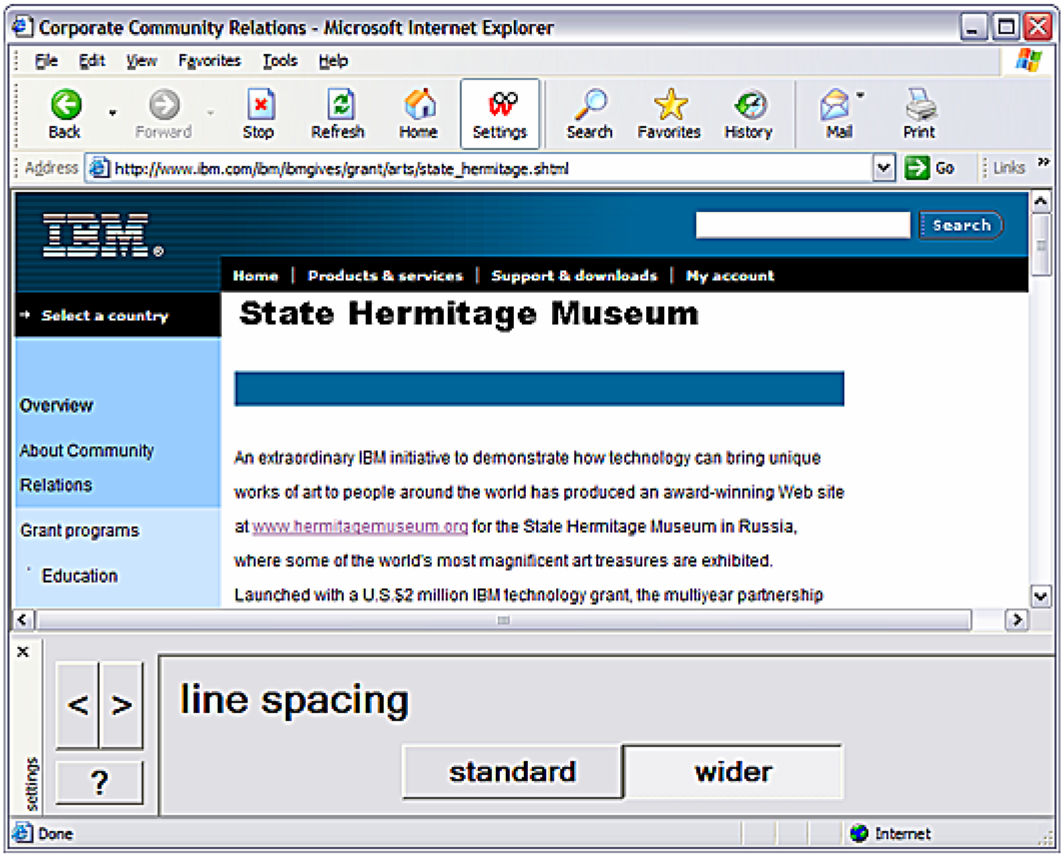

Figure 4. The same Web page as in Figure 3, but with the 'wider' line spacing transformation applied.

persons with disabilities contacted us, wishing to use the software. At this point in time, we find that the software is used not only by organizations serving older adults, but also by organizations serving persons with visual and dexterity disabilities, students with learning disabilities or developmental delays, and speakers of English as a second language. The software has also been translated into six languages and used by organizations serving similar communities in countries other than the United States. We now have a few thousand users of the software. We report here on summary data for these users. More detailed presentations of the data are available elsewhere [7], [8].

Across this full spectrum of abilities, the speak text adaptation is used by approximately 41% of our users. This feature utilizes IBM's ViaVoice® text-to-speech capability. The implementation of this speak text feature differs from the kind of text-to-speech support used in screen readers for blind users [1], [22] in that users point the mouse at that portion of the page they wish to have read aloud. As the mouse hovers over text, links, or images, the corresponding text is read aloud (for images, the ALT text that is present is spoken). This feature is particularly popular among low vision users who use it as reinforcement when reading. It is also popular among low literacy individuals, along with the ability to have individual words read aloud upon a double click.

Also selected by approximately 41% of our users is text size enlargement. Most of these users choose our large and larger sizes, with fewer than 2% choosing our largest size. Similarly, while the page magnification is activated by about 32% of our users, it is predominantly the smallest magnification that is used. It, thus, appears that the majority of our users found this amount of text and page enlargement sufficient to make their Web browsing more comfortable.

Used nearly as often as these features are text style and color contrasts, which were each selected by more than 30% of our users. The transformations for banner text, image enlargement, and enlargement of browser controls are employed by approximately a quarter of our users. Only about 10% - 20% of our users take advantage of the line spacing, letter spacing, and stop animation features. Hiding images and backgrounds were employed by only 10% of our users.

Information regarding the setting of the keyboard options based on monitoring of typing is currently underway [18].

Our most recent page transformation is the 'one column' feature and we had only a few users who had this option available to them. As a result, we are not reporting usage statistics at this time. With this option, multi-column pages are transformed into one column. As shown in Figure 5, when Web page authors provide skip navigation links, our page linearization feature brings the user directly to the main page content. Critically, no content from the page is eliminated with this transformation.

6. ACCESSIBILITY AND USABILITY

The software we described here is complementary in approach to accessible Web page design as prescribed in guidelines. The majority of Web audits and checklists focus on the content

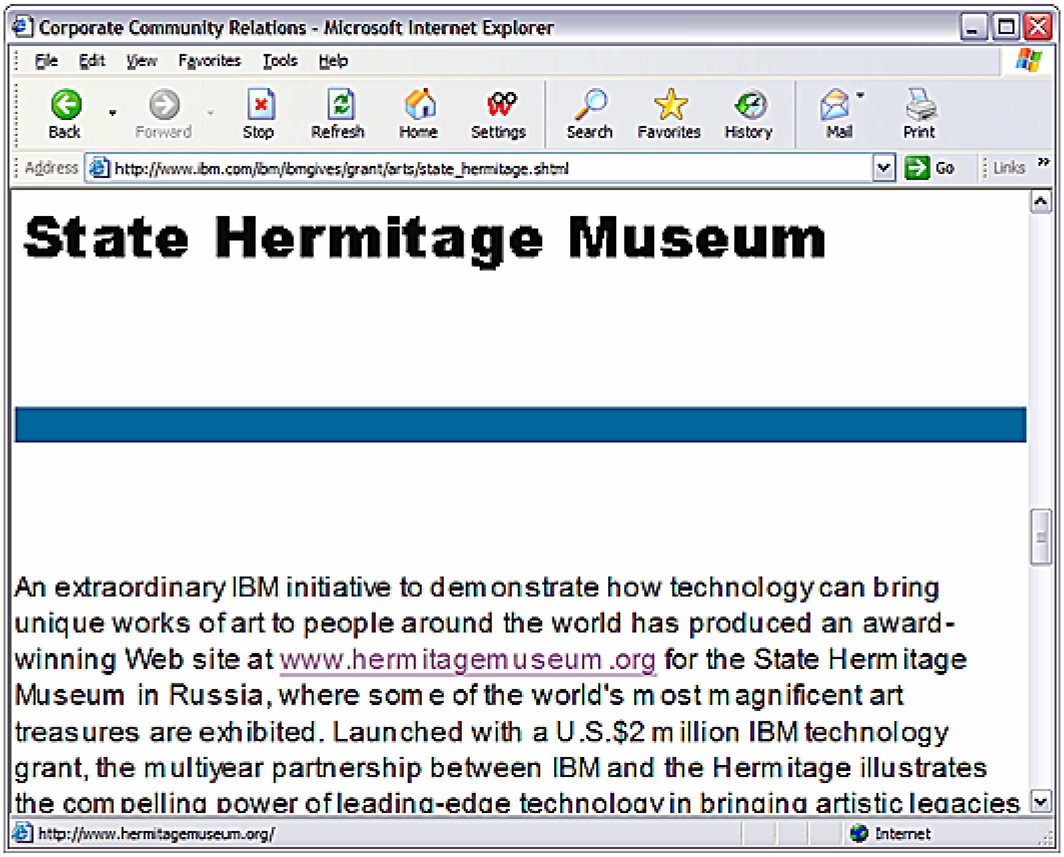

Figure 5. The same Web page as in Figure 3, but with 175% page magnification and the 'one column' transformation applied.

development standards of the W3C [21] or the technical performance standards of Section 508 [14]. These standards focus on what must be done by content providers when creating pages so that pages will be accessible. In many cases, these standards do not address making Web pages directly usable by people, but rather address making Web pages capable of being rendered by assistive technology devices.

Content providers are encouraged to provide text-only shadow pages to meet functional performance standards or when technical performance standards cannot be met. There are downsides to this alternative, however. Creating shadow pages is costly and there are concerns about whether shadow pages are updated as often as their original counterpart. The biggest downside, however, is that such an alternative does not adequately address the diversity of user needs. For example, a text-only shadow site cannot provide for the variety of content enlargements and color changes that different users require. Moreover, the lack of images makes some users feel they are missing important information and, in some cases, can even make processing of the page *more* difficult, for example, as with learning disabled users who may rely on images for context. In order to address this diversity, it is necessary to provide for user control over content delivery such that individuals are able to make adjustments to pages, tailored to their owns needs.

The software described in this paper provides a cost effective means for meeting the needs of the diverse population of users whose needs are not fully addressed by content accessibility standards [13]. It does not address the technical standards for accessibility, but would more properly fall into the category of making Web pages more usable. Accessible pages are not necessarily usable pages [12]. The Web pages used as examples in this paper were fully compliant with accessibility standards. Despite this, these pages can be difficult for persons due to various limitations they possess. As discussed, our technology allows people to make a number of adaptations that can greatly increase their ease of use for Web pages, without putting additional requirements on Web content providers. The types of transformations made by our technology are consistent with guidelines that focus on functional performance standards [15]. They are also consistent with the concept of user control of presentation proposed by the User Agent Accessibility Guidelines [20].

7. SUMMARY AND CONCLUSIONS

The client architecture presented here successfully addresses the problems inherent in our previous proxy server design. It eliminated the issues of secure Web sites, setting the proxy, and copyright. The use of the browser's own parsing to create the DOM eliminated problems with accurately rendering Web pages. Problems of scale also were eliminated in that transformations are now made on individual client machines, eliminating the need for large and powerful servers to perform multiple renderings and transformations simultaneously. The browser and operating systems transformations and adaptations are performed by the client without any delay in processing. The DOM transformations do require processing, however, and may introduce some delay in page presentation, depending on the speed of the client. On older, slower machines, the delay may be perceptible for complex Web pages. In most cases, however, the delay is negligible.

This is not the first piece of software designed to transform Web page presentation. The software is unique, however, in the range of transformations and adaptations that users can adopt. It is clear that the phrase "one size fits all" does not apply to those we support. Our users have complex, interacting, and changing abilities. Our software provides the ability for them to easily apply multiple transformations and adaptations, in any combination depending on their immediate needs, to make Web pages more usable.

8. ACKNOWLEDGMENTS

We would like to acknowledge the contributions of our colleagues at IBM's Watson Research Center, Jonathan Brezin, Susan Crayne, Peter Fairweather, Cal Swart, Beth Tibbitts, and Shari Trewin, and in other IBM locations, Sam Detweiler and Rich Schwerdtfeger.

9. REFERENCES

[1] Asakawa, C., & Itoh, T. User interface of a Home Page Reader. In *Proceedings of ASSETS'98* (Marina del Rey, CA, April, 1998), ACM Press, 149-156.

[2] Asakawa, C., & Takagi H. Annotation-based transcoding for nonvisual Web access. In *Proceedings of ASSETS'00* (Arlington, VA, November, 2000), ACM Press, 172-179.

[3] *Creating Senior-Friendly Web Sites*. Center for Medicare Education. Available at http://www.medicareed.org/pdfs/ibv1n4.pdf

[4] Czaja, S. J., & Lee, C. C. (2003). Designing computer systems for older adults. In J. Jacko & A. Sears (eds) *The human-computer interaction handbook*, Erlbaum, Mahwah, NJ, 2003, 413-427.

[5] Fairweather, P.G., Hanson, V.L., Detweiler, S.R., and Schwerdtfeger, R.S. From Assistive Technology to a Web Accessibility Service. In *Proceedings of ASSETS'02* (Edinburgh Scotland, July 2002), ACM Press, 4 – 8.

[6] Gregor, P., & Newell, A. F. Designing for dynamic diversity – making accessible interfaces older people. In *Proceedings of the Workshop on Universal on Accessibility of Ubiquitous Computing, WUAUC'01* (Alcácer do Sal, Portugal, May, 2000), ACM Press, 90-92.

[7] Hanson, V. L., & Crayne, S. Personalization of Web browsing: Adaptations to meet the needs of older adults. Manuscript submitted for publication.

[8] Hanson, V. L., & Richards, J. T. Achieving an accessible World Wide Web. *Behaviour and Information Technology*, in press.

[9] *Header field definitions. Section 14.9.5 No-transform directive*. http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9.5

[10] *Making Your Web Site Senior Friendly: A checklist. National Institute on Aging*. Available at http://www.nlm.nih.gov/pubs/checklist.pdf

[11] *National Institute on Aging. Older Adults and Information Technology: A Compendium of Scientific Research and Web Site Accessibility Guidelines*. Bethesda, MD: National Institute on Aging, 2002.

[12] Powlik, J. J., & Karshmer, A. I. When accessibility meets usability. *Universal Access in the Information Society*, 2002, 1, 217-222.

[13] Richards, J. T., & Hanson, V. L. Web accessibility: A broader view. In *Proceedings of the Thirteenth International ACM World Wide Web Conference, WWW2004* (New York City, NY, May, 2004), ACM Press, 72-79.

[14] *Section 508: The Road to Accessibility*. http://www.section508.gov/

[15] *Summary of Section 508: Functional Performance Criteria (Subpart C)*. Available at http://www.section508.gov/index.cfm?FuseAction=Content&ID=11#functional

[16] *SeniorNet*. http://www.seniornet.org

[17] Takagi, H., Asakawa, C., Fukuda, K., & Maeda, J. Site-wide annotation: reconstructing existing pages to be accessible. In *Proceedings of ASSETS'02* (Edinburgh, Scotland, July, 2002), ACM Press, 81 – 88.

[18] Trewin, S. Automating accessibility: The dynamic keyboard. In *Proceedings of ASSETS'04* (Atlanta, GA, October, 2004), ACM Press.

[19] Trewin, S., & Pain, H. A model of keyboard configuration requirements. In *Proceedings of ASSETS'98* (Marina del Rey, CA, April, 1998), ACM Press, 173 - 181.

[20] *User Agent Accessibility Guidelines*. Available at http://www.w3.org/TR/2002/PR-UAAG10-20021016/

[21] *W3C Web content accessibility guidelines*. Available at http://www.w3.org/WAI/

[22] Zajicek, M., Powell, C., & Reeves, C. A Web navigation tool for the blind. In *Proceedings of ASSETS'98* (Marina del Rey, CA, April, 1998), ACM Press, 204-220.