

WebAnywhere: A Screen Reader On-the-Go

Jeffrey P. Bigham, Craig M. Prince and Richard E. Ladner
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195 USA
{jbigham, cmprince, ladner}@cs.washington.edu

ABSTRACT

People often use computers other than their own to access web content, but blind users are restricted to using only computers equipped with expensive, special-purpose screen reading programs that they use to access the web. *WebAnywhere* is a web-based, self-voicing web browser that enables blind web users to access the web from almost any computer that can produce sound without installing new software. The system could serve as a convenient, low-cost solution for blind users on-the-go, for blind users unable to afford a full screen reader and for web developers targeting accessible design. This paper overviews existing solutions for mobile web access for blind users and presents the design of the WebAnywhere system. WebAnywhere generates speech remotely and uses prefetching strategies designed to reduce perceived latency. A user evaluation of the system is presented showing that blind users can use WebAnywhere to complete tasks representative of what users might want to complete on computers that are not their own. A survey of public computer terminals shows that WebAnywhere can run on most.

Categories and Subject Descriptors

K.4.2 [Social Issues]: Assistive technologies for persons with disabilities; H.5.2 [Information Interfaces and Presentation]: User Interfaces

General Terms

Design, Human Factors, Experimentation

Keywords

Screen Reader, Web Accessibility, WebAnywhere, Blind Users

1. INTRODUCTION

This paper introduces *WebAnywhere*, a web-based screen reader that can be used by blind individuals to access the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WAA2008 - Technical, April 21-22, 2008, Beijing, China. Co-Located with the 17th International World Wide Web Conference.
Copyright 2008 ACM ...\$5.00.



Figure 1: People often use computers besides their own, such as computers in university labs, library kiosks, or friends' laptops.

web from almost any computer that has both an Internet connection and audio output. Blind computer users employ software programs called screen readers to convert the visual interfaces and information on the screen to speech. These programs also provide shortcut keys that make it feasible to use a computer non-visually and without a mouse. Worldwide there are more than 38 million blind individuals (more than 1 million in the United States) whose access to web resources relies on acquiring the use of a screen reader [32].

Screen readers are expensive, costing nearly a thousand dollars for each installation because of their complexity, relatively small market and high support costs. Development of these programs is complex because the interface with each supported program must be deciphered independently. As a result of their expense, screen readers are not installed on most computers, leaving blind users on-the-go unable to access the web from any computer that they happen to have access to and many blind users unable to afford a screen reader unable to access the web at all. Other portable screen readers require permission to install new software, which users often do not have on public computers. WebAnywhere can provide web access for free from most computers.

Even blind web users who have a screen reader installed at home or at work face difficulty accessing the web everywhere that sighted people can. From terminals in public libraries to the local gym, from coffee shops to pay-per-use computers at the airport, from a friend's laptop to a school laboratory;

computers are used for a variety of useful tasks that most of us take for granted, such as checking our email, viewing the bus schedule or finding a restaurant. Blind people cannot access the web from many remote locations even if they are accustomed to using a computer at home using a screen reader. Few would argue that the ease of use of web mail or document editors has surpassed desktop analogs, but their popularity is increasing, indicating the rising importance of accessing the web from wherever someone happens to be. A primary advantage of web applications is anywhere access to your information, but this is less exciting if you will always be using the same computer to access them.

The cost of screen readers is a barrier for web access for potential screen reader users. Libraries are known to provide an invaluable connection to web information for those with low incomes [6], and, while almost all libraries in the United States today provide Internet access, many do not provide screen readers because of their expense. Even when they do, they are provided on a limited number of computers. Most blind people with low incomes cannot afford a computer and a screen reader of their own, and governmental assistance in receiving one (in countries where such programs exist) are often tied to a demonstrated need for one because of school or employment, as it is in the United States. Unemployed blind people often miss out on such services even though they could potentially benefit the most. In countries where these programs do not exist, the problem may be worse.

For blind users unable to afford a full screen reader, Web-Anywhere might serve as a temporary alternative. Voice output while navigating through a web page can also be beneficial for people who have low vision or dyslexia. Web developers have been shown to produce more accessible content when they have access to a screen reader [19], but can be deterred from using one by their expense and the hassle of installing one. All of these users could use WebAnywhere for free from anywhere.

In this paper, we explore the design of the WebAnywhere system and our approach to making it usable by blind web users on most available computers. To enable WebAnywhere to work on as many computers as possible, speech is generated remotely and then transferred to the client. We investigate the usability concerns of the resulting latency and briefly explore the steps taken to minimize it. We demonstrate that WebAnywhere works on most publicly-available terminals and in the process provide a summary of the characteristics of existing public terminals that may help future designers of accessible web applications that want their applications to be accessible on most computers. Finally, we present a user evaluation that demonstrates that blind users can use WebAnywhere to access the web.

2. RELATED WORK

Many existing solutions convert web content to voice to enable access by blind individuals. Three important dimensions on which to compare them are functionality, portability and cost. WebAnywhere provides full web-browsing functionality from any computer for free to users.

2.0.1 Screen Readers

Screen readers, such as JAWS [17] or Window-Eyes [35], are special-purpose software programs that cost more than \$1,000 US. The Linux Screen Reader [18], the Orca screen

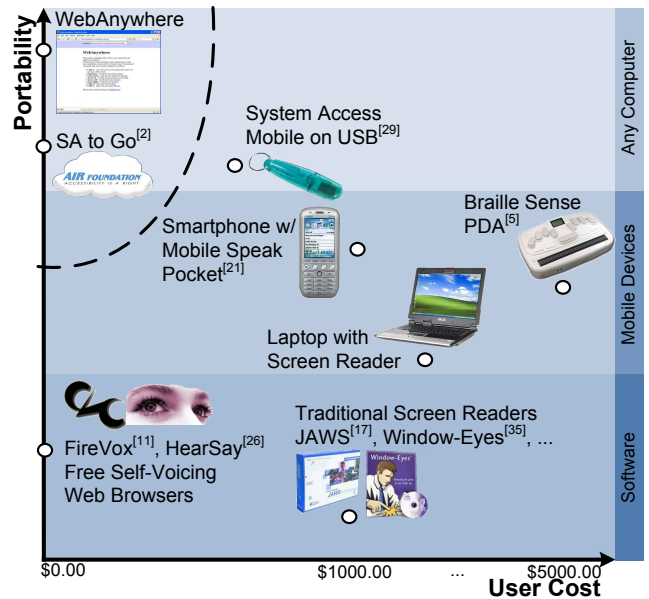


Figure 2: Many products enable web access to blind individuals but few have high availability and low cost (upper-left portion of this diagram). Only systems that voice both web information and provide an interface for browsing it are included.

reader [24] for the GNOME platform and the NVDA screen reader for Windows [23] are free alternatives to the commercial products. Screen readers are seldom installed on computers not normally used by blind individuals because of their expense and because their owners are unaware of free alternatives or that blind users might want to use them. Fire Vox is a free extension to the Firefox web browser that provides screen reading functionality [11], the HearSay web browser is a standalone self-voicing web browser [26], and aiBrowser is a self-voicing web browser that targets making multimedia web content accessible [20]. Although free, these alternatives are similarly unlikely to be installed on most systems.

Users are rarely allowed to install new software on public terminals, and installing software takes time and may be difficult for blind users without using a screen reader. Many users would also be hesitant to install new software on a friend’s laptop. WebAnywhere is designed to replicate the web functionality of screen readers in a way that can be easily accessed from any computer, requires minimal permissions on the client computer, and starts up quickly without requiring a large download before the system can be used.

Recent versions of Macintosh OS X include a screen reader called VoiceOver which voices the contents of a web page and provides support for navigating through web content [33]. Most computers do not run OS X, and, on public terminals, access to features not explicitly allowed by an administrator may be restricted. Windows XP and Vista include a limited-functionality screen reader called Narrator, which is described as a “text-to-speech utility” and does not support the interaction required for web browsing [36].

2.0.2 Mobile Alternatives

Mobile access alternatives can be quite costly. PDA solutions can access the web in remote locations that offer

wireless Internet and usually also offer an integrated Braille display that can be used on any computer. The GW Micro Braille Sense [5] costs roughly \$5,000 US but also offers an integrated Braille display. A Pocket PC device and the screen reading software designed for it, Mobile Speak Pocket [21], together cost about \$1,000 US. Many cannot afford or would prefer not to carry such expensive devices.

The Serotek System Access Mobile (SAM) [29] is a screen reader designed to run on Windows computers from a USB key without prior installation. It is available for \$500 US and requires access to a USB port and permission to run arbitrary executables. The Serotek System Access To Go (SA to Go) screen reader can be downloaded from the web via a speech-enabled web page, but the program requires Windows, Internet Explorer, and permission to run executables on the computer. The AIR Foundation [2] has recently made this product free. A self-voicing Flash interface for downloading and running the screen reader is provided.

2.0.3 Alternatives with Limited Functionality

Some web pages voice their own content, but are limited either by the scope of information that can be voiced or by the lack of an accessible interface for reaching the speech. Talklets enable web developers to provide a spoken version of their web pages by including code that acts with the Talklet server to play a spoken version of each page [30]. This speech plays as a single file and neither its playback nor the interface of the page can be manipulated by users. Scribd.com converts documents to speech [28], but the speech is available only as a single MP3 file that does not support interactive navigation. The interface for converting a document is also not voiced. The National Association for the Blind, India, provides access to a portion of their web content via a self-voicing Flash Movie [22]. The information contained in the separate Flash movie is not nearly as comprehensive of the entire web page, which could be read by WebAnywhere.

Web information can also be retrieved using a phone. The free GOOG-411 service enables users to access the business information from Google Maps using a voice interface over the phone [14]. For a fee, email2phone provides voice access to email over the phone [13].

2.0.4 Summary

Products that provide full web-browsing functionality are shown in Figure 2. The portability axis is approximate. Solutions that can be run on any computer can also be run on wireless devices and are therefore rated more highly. Mobile phones are more portable than other solutions but only when cellphone service is available. WebAnywhere will be able to run on many mobile devices, regardless of the underlying platform, as they are increasingly supporting more complex web browsers that can play sound. The WebAnywhere web application is more highly portable than the Serotek System Access Mobile, which can run only on Windows computers on which users have permission to run arbitrary executables. Braille Sense PDAs use a proprietary operating system and some versions cannot connect to wireless networks using WPA security. WebAnywhere is designed to be free and highly-available, but other solutions may be more appropriate or provide more functionality for users with different requirements using different devices.

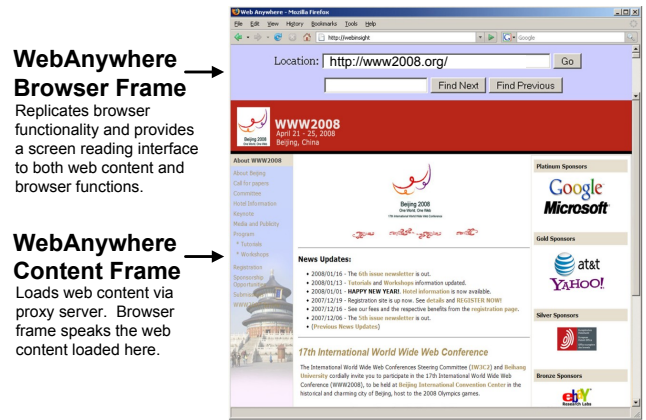


Figure 3: WebAnywhere is a self-voicing web browser inside a web browser.

2.1 Accessibility Evaluation

Web developers can use WebAnywhere to help them create more accessible content. Mankoff *et al.* showed that web developers create more accessible web pages when they review them with a screen reader [19]. WebAIM provides an online, interactive Flash movie that demonstrates what it is like to access the web using a screen reader in an effort to improve the accessibility of the sites produced by web developers [34]. WebAnywhere is a convenient, inexpensive solution that enables developers to realistically evaluate the accessibility of their own web pages using a screen reader.

3. WEBANYWHERE DESIGN

Web sites employing Web 2.0 technology are quickly gaining popularity but are not attractive to blind users. The dynamic page elements used in such sites are often inaccessible to blind users using existing screen readers. The interfaces provided in web applications are not as good as their desktop equivalents, but are used because they can be accessed from anywhere. Blind users generally only use their own computers to access the web because most computers do not have a screen reader installed on them. WebAnywhere was designed to enable blind users to use the web from anywhere and interacts directly with the Document Object Model (DOM) of the page to provide better access to dynamic pages.

To use the system, users first browse to its web page. WebAnywhere speaks both its interface and the content of the web page that is currently loaded (initially a welcome page). Users can navigate using this web page to other web pages and the WebAnywhere interface will enable new pages to be spoken to the user. No separate software needs to be downloaded or run by the user; the system runs entirely as a web application with minimal permissions (Figure 3).

3.1 User Interaction

WebAnywhere includes much of the functionality included in existing screen readers for enabling users to interact with web pages. The system traverses the DOM of each loaded web page using a pre-order Depth First Search (DFS), which is approximately top-to-bottom, left-to-right in the visual page. As it reaches each element of the DOM, it speaks the element's textual representation to the user. For instance,

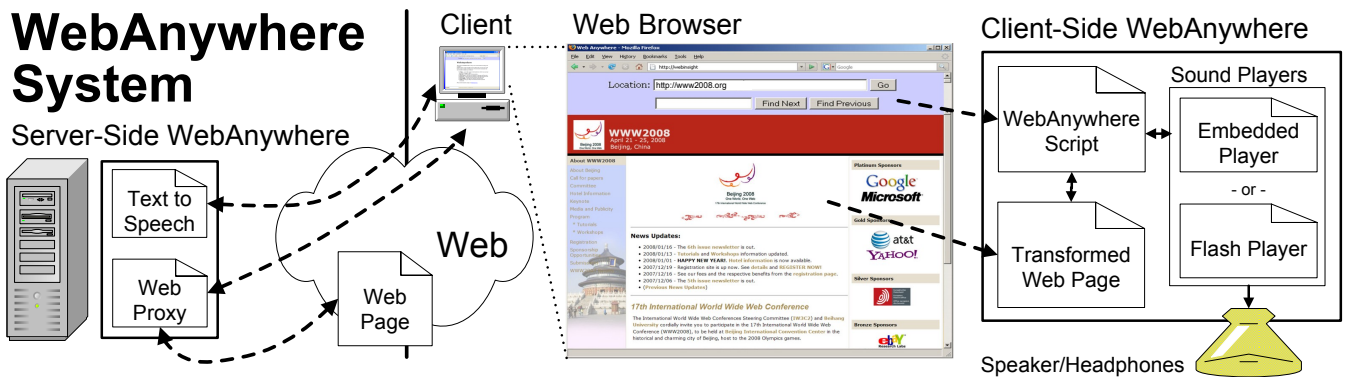


Figure 4: The WebAnywhere system consists of server-side components that convert text to speech and proxy web content; and client-side components that provide the user interface, coordinate which sound will be played and play the sounds.

upon reaching a link containing the text “Google,” it will speak “Link Google.” Users can skip forward or backward in this content by sentence, word or character. Users can also skip through page content by headings, input elements, links, paragraphs and tables. They can navigate through tables by moving cell-by-cell by either row or column. In input fields, the system speaks the characters typed by the user and allows them to review what they have typed. Figure 5 displays a subset of the keyboard shortcuts offered by WebAnywhere.

Most existing screen readers differentiate between a forms mode in which users can interact with form elements on the page and a browse mode in which users can browse the content of the page. Screen readers originally did not have direct access to the DOM of web pages that they were reading and, instead, created an off-screen model of them that was read instead. WebAnywhere has direct access to the DOM of the page and, therefore, does not need to support these different modes. This direct access also allows WebAnywhere to support dynamic page changes better than many existing screen readers which often fail to update their off-screen models when page content changes.

As a web application running with minimal permissions, WebAnywhere does not have access to the interface of the browser. It instead replicates the functionality already provided, such as providing its own address bar where the URL of the current page is displayed and users can type to navigate to a new page (Figure 3).

3.2 User-Driven Design

WebAnywhere was designed with close consultation with blind users. These potential users of WebAnywhere have offered valuable feedback on its design.

During preliminary evaluations with 3 blind web users, participants wanted to be able to customize the shortcut keys used and other features of WebAnywhere in order to emulate their usual screen readers (either JAWS or Window-Eyes) [10]. To support this, WebAnywhere now includes user-configurable components that specify which key combinations to use for various functionality and the text that is read when specified actions are taken. Users can also choose to emulate their preferred screen reader (either JAWS or Window-Eyes) and their preferred browser (Internet Explorer or Firefox). For instance, Internet Explorer announces

“Type the Internet address of a document or folder, and Internet Explorer will open it for you” when users move to the location bar, while users will hear “Location Bar Combo Box” if they are using Firefox. These settings can be changed using a web-based form and are saved for each individual user.

To load their personal settings, users first press a shortcut key. The system asks them to enter the name of their profile and press enter when they are finished. It then applies the appropriate settings. Users can create an account and edit their personal settings either using a screen reader to which they are already accustomed or by using the default interface provided by WebAnywhere. Explanations of the available functionality and initial keyboard shortcuts assigned to each are read when the system first loads. It is unclear to what extent frequent users will want to use personalized settings, but the option to use personalized settings may ease the transition to WebAnywhere from another screen reader. In the future, personal profiles may also enable users to specify their preferred speech rate and interface language.

When browsing the web, screen readers use an off-screen model of each web page that is encountered, which results in the screen reader exposing two different complementary but incomplete modes of operation to the user. Because WebAnywhere accesses the DOM of the web page directly, it does not need a separate forms mode. This has the advantage of immediately updating when content in the page dynamically changes. Traditional screen readers must be manually toggled between a “forms mode” and “browse mode”¹ using a specific control key. Even though this is cumbersome and unnecessary in WebAnywhere, it caused confusion for users that were accustomed to it, and so this behavior can be emulated in WebAnywhere.

Our consultants felt that the main limitation of the system was the limited functionality that provided to skip through page content relative to other screen readers. Users felt that WebAnywhere most needed the following two features:

- **Skipping functionality** - Users have become accustomed to the rich array of shortcuts designed to enable users to skip through content. Our initial system only supported skipping by focusable element using

¹Names for these modes differ; these are used by JAWS.

the TAB key, but most screen readers provide shortcuts to skip through web content by heading, by input element, by paragraph and by link.

- **Find feature** - Users wanted to be able to search for text in the page using the familiar find functionality provided by most browsers, which has been shown valuable for blind web users [12]. The existing find functionality in the web browser is not accessible using WebAnywhere because, as a web application, WebAnywhere can only access the elements in the web pages it loads.

We used these preliminary evaluations to direct the priorities for development of WebAnywhere and the system now includes these features. Individuals also wanted a variety of other functionality available in existing screen readers, such as the ability to spell out the word that was just spoken, to speak user-defined pronunciations for words, and to specify the speech rate. These can be implemented in future versions of WebAnywhere.

3.3 Reaching WebAnywhere

Before using WebAnywhere, blind users must first navigate to the WebAnywhere web page. Blind users have proven adept at using computers to start a screen reader when one is not yet running. For instance, some screen readers required users to login to the operating system without using the screen reader. Existing solutions, such as the Serotek System Access Mobile [29], share this requirement and are still used. In most locations, blind users can ask for assistance in navigating to the WebAnywhere page and then browse independently. This issue is explored more in our survey of public terminals presented in Section 5.

Windows provides a limited voice interface that could be used to reach WebAnywhere. Windows Narrator can be started using a standard shortcut key and can voice the run dialog enabling users to navigate to the WebAnywhere URL. Web Narrator’s rudimentary support for browsing the web is not sufficient for web access, but would enable users to open the WebAnywhere web page.

4. WEBANYWHERE SYSTEM

WebAnywhere is designed to function on most computers that have Internet access and that can play sound. To this end, its design has carefully considered independence from a particular web browser or plugin. To facilitate its use on public systems with varying capabilities and on which users may not have permission to install arbitrary components, functionality that would require these have been moved to a remote server. The web application can also play sound by using several sound players available in different web browsers.

The system consists of the following three components (Figure 4): (i) client-side Javascript, which supports user interaction, decides which sounds to play and interfaces with sound players to play them, (ii) server-side text-to-speech generation and caching, and (iii) a server-side transformation proxy that makes web pages appear to come from a local server to overcome cross-site scripting restrictions.

4.1 WebAnywhere Script

The client-side portion of WebAnywhere is a self-voicing web browser that can be run inside an existing web browser

Navigation Functionality	
CTRL + l	Focus location text field.
CTRL + f	Focus finder text field.
Reading Granularity	
↑ or ↓	Read next/previous element.
→ or ←	Read next/previous word.
SHIFT + → or ←	Read next/previous character.
Skipping Functionality	
TAB	Skip to next focusable element.
CTRL + h	Skip to next heading.
CTRL + i	Skip to next input element.
CTRL + r	Move to next row in table.
CTRL + d	Move to next column in table.

* Pressing SHIFT in combination with these keys will reverse skipping direction.

Figure 5: Selected shortcut functionality provided by WebAnywhere and the default keys assigned to each. The system implements the functionality for more than 30 different shortcut keys. Users can customize the keys assigned to each.

(Figure 3). The system is written in cross-browser Javascript that is downloaded from the server, allowing it to be run in most modern web browsers, including Firefox, Internet Explorer and Safari. The system captures all key events, allowing it to maintain control of the browser window. WebAnywhere’s use of Javascript to capture a rich set of user interaction is similar to that of UsaProxy [4] and Google Analytics [15], which are used to gather web usage statistics. To enable the Javascript to have access to the DOM of the pages that it loads, these pages are loaded through a modified version of the web-based proxy PHPProxy [3].

4.2 Producing and Playing Speech

Speech is produced on the server using the free Festival Text-to-Speech System [31]. The sounds produced are converted to the MP3 format because this format can be played by most sound players available in browsers and creates small files necessary for maintaining low latency. For example, the sound “Welcome to WebAnywhere,” played when the system loads, is approximately 10k, while the single letter “t”, played when users type the letter, is 3k. Sounds are cached on both the server so they don’t need to be generated again when they are requested again and on the client as part of the existing browser cache.

The primary mechanism for playing sound is using Adobe Flash via SoundManager 2 Flash Object [27]. This Flash object provides a Javascript bridge between the WebAnywhere Javascript code and the Flash sound player. It provides an event-based API for sound playback that signals when sounds have started playing and when they have finished. It is also able to begin playing sounds before they have finished

downloading, resulting in low perceived latency. Adobe reports that version 8 or later of the Flash player, required for full sound support using Flash, is installed on 98.5% of computers [1].

WebAnywhere also supports embedded sound players, such as Quicktime and Windows Media Player. The timing of sounds played using these players is more difficult to control because of their limited Javascript interface, which cannot fire events when sounds have finished playing. When this method of playback is used, users manually advance to the next sound using a keyboard shortcut and cannot have web pages read to them automatically from start to finish. Most users prefer to control navigation manually even when not using this player. Flash is installed on most public terminals, but support for embedded players provides access to WebAnywhere even when it is not.

4.3 Maintaining Focus

WebAnywhere runs as a web application in the host web browser and does not have access to windows that are not part of its display. Both the browser and the operating system can take focus from WebAnywhere and because the system cannot access these components the user will lose control. WebAnywhere uses several strategies to both prevent the browser from losing focus in the first place and guiding the user back to the system when focus is lost.

WebAnywhere attempts to prevent web pages from causing WebAnywhere to lose focus. It aggressively blocks popup windows and page redirects that are not sent through the system's proxy. The Javascript code used in many pages to ensure that they are displayed in the top-level window are rewritten so that the WebAnywhere content frame is the top-level window.

The browser and other programs can still cause the WebAnywhere window to lose focus, and the system is designed to guide the user back to the system. The focus of WebAnywhere can be displaced by either modal dialog boxes or by other windows. The former can be dismissed by either pressing the escape key or by pressing enter; the latter by switching windows until WebAnywhere regains focus. The system will announce itself when it regains focus. Upon losing focus, the system says the following to help users return its focus: "The system has lost focus. Try pressing Escape and then ALT-TAB until WebAnywhere announces itself."

4.4 Limitations

WebAnywhere does not currently work on pages that use frames and users cannot interact with Adobe Flash objects on web pages. By retrieving web content through the web proxy, all content appears to come from the same domain, violating the *same origin* security policy implemented by web browsers. This enables the cross-site scripting necessary for allowing WebAnywhere to read and control web pages located anywhere on the web, but could also be used by scripts with malicious intent to violate the security of downloaded content. We can address these limitations in future versions of WebAnywhere.

Existing screen readers provide a voice interface for programs other than the web browser, but WebAnywhere is not able to support this functionality. Web access dominates the functionality that people want to use on computers that are not their own, but people often want to read document files that are normally handled by external viewers,

Public Terminal Survey

	WebAnywhere Usable	Screen Reader	Help Available	Flash Installed	Other Installed	Sound Player	Sound Initially Audible	Headphone Jack	Headphones	Speakers
Internet Cafes (3)	3	0	3	3	3	3	3	2	1	1
Kiosks (3)	2	0	2	2	3	2	2	3	0	2
Libraries (5)	5	0	5	5	4	3	3	5	1	1
Other (4)	4	0	2	4	3	4	4	3	0	1
All (15)	14	0	12	14	13	12	12	13	2	5

Figure 7: Survey of public computer terminals by category indicating that WebAnywhere can run on most of them.

which WebAnywhere cannot read directly. WebAnywhere redirects links pointing to Adobe PDF files to the HTML version provided by Google, which enables users to browse PDF files when they are in Google's cache. Future versions may support redirecting supported files to appropriate web viewers, such as Google Documents, which can open many document formats such as Microsoft Word documents [16].

5. PUBLIC COMPUTER TERMINALS

WebAnywhere is designed to operate on public terminals, but its ability to do so is dependent on the technical capabilities of such computer terminals. In the United States, nearly all public libraries provide Internet access to their patrons, and 63% of these libraries provide high speed connections [6]. As of 2006, 22.4% of libraries specifically provided Internet-based video content and 32.8% provided Internet-based audio content [6]. Other public terminals likely have different capabilities. The aspects of public terminals explored here will also determine the ability of other web applications to enable self-voicing technology, and is, therefore, generally applicable beyond WebAnywhere.

We surveyed 15 public computer terminals available in Seattle, WA, to get an idea of their technical capabilities and elements of the environment where they were located that could influence how WebAnywhere would be used. We visited computers in the following locations: 5 libraries, 3 Internet cafes, 3 university & community college labs, 2 business centers, a gym and a retirement community. The computers were all managed by different groups. For instance, we visited only one Seattle public library. Although we considered only public terminals located in Seattle, we found public terminals with diverse capabilities, suggesting that our results may generalize. For instance, while most ran Windows XP as their operating system, two ran Macintosh OS X, and one ran Linux. The terminals were designed for different use cases. Several assumed users would stand while accessing them, one was used primarily as a print station and many appeared to be several years old.

Figure 7 summarizes the results of this survey. Most computers tested were capable of running WebAnywhere, and in 12 of the locations, someone was available in an official capacity to assist users. In all locations, people (including employees) were nearby and could have also assisted users.

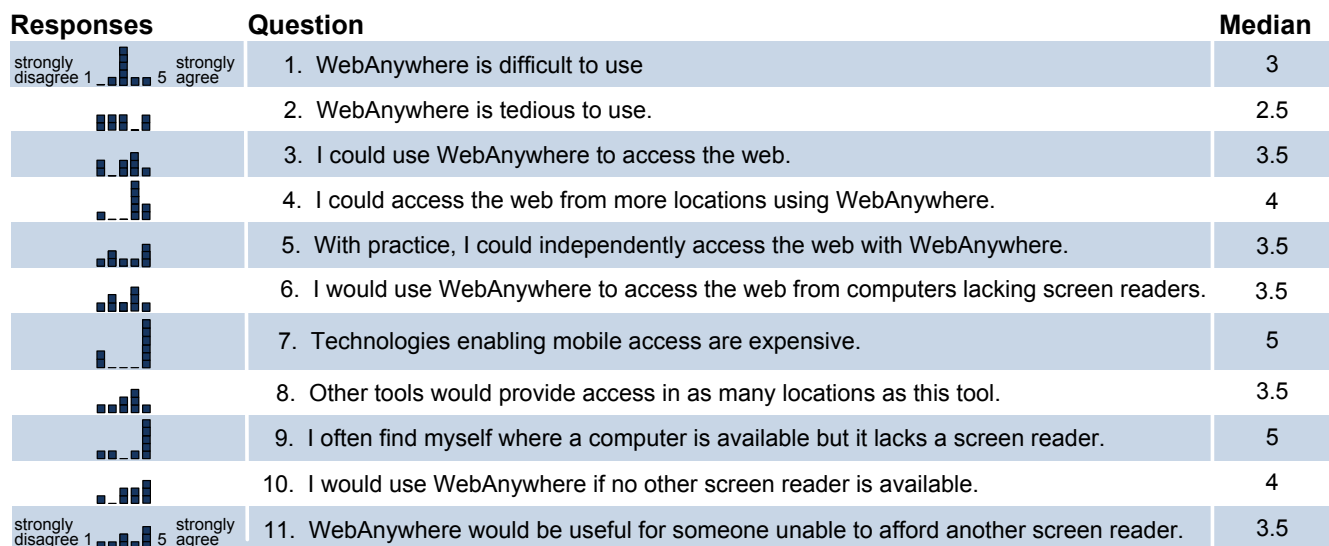


Figure 6: Participant response to the WebAnywhere, indicating that they saw a need for a low-cost, highly-available screen-reading solution (7,8,9) and thought that WebAnywhere could provide it (3,4,6,10). Full results available in Appendix A.

The most notable restriction to access that we found was that only 2 locations (a library and an Internet cafe) provided headphones for users. However, in all of the libraries that we visited, we were able to ask to use headphones. We assumed users would bring headphones, and felt this was a reasonable requirement given that a simple set of headphones are inexpensive and many people carry them already for listening to their music players and other devices. In 5 locations that did not provide headphones, speakers were available. Using speakers is not ideal because it renders a user’s private browsing public and could be bothersome to others, but at least in these locations users could access the web without requiring them to bring anything with them. One location was restricted to using the embedded sound player, which suggests that while supporting it could help in some cases, access on most computers could probably be achieved only using the Flash player to play sound. On 14 out of 15 systems, blind users could potentially access web content even though none of the computers had a screen reader installed on them.

6. USER EVALUATION

In order to investigate the usability and perceived value of WebAnywhere, we conducted a study with 8 blind users (4 female) ranging in age from 18 to 51. Participants represented a diversity of screen reader users, including both students and professionals, involved in fields ranging from the sciences and law to journalism and sales. Their self-described, screen-reader expertise varied from beginner to expert. Similarly, their experience using methods for accessing the web when away from their own computers using the methods described in Section 2 varied considerably. Participants were compensated with \$15 US for their participation in our study, and none had participated in earlier stages of the development of WebAnywhere.

Two of our participants were located remotely. Remote user studies can be particularly appropriate for users with disabilities for whom it may be difficult or costly to con-

duct in-person studies. Such studies have been shown to yield similar quantitative results, although risk collecting less-informative qualitative feedback [25]. We conducted interviews with remote participants to gather valuable qualitative feedback.

We examined (i) the effects of technological differences between our remote screen reader and a local one, and (ii) the likelihood that participants would use WebAnywhere in the future. While we did not explicitly compare the screen reading interface with existing screen readers, all of our participants had previously used a screen reader and many of their comments were made with respect to this experience.

In this evaluation, participants were first introduced to the system and then asked to browse the WebAnywhere homepage using it. They then independently completed the following four tasks: searching Google to find the phone number of a local restaurant, finding when the next bus will be arriving at a bus stop, checking a Gmail email account, and completing a survey about their experience using WebAnywhere. Gmail.com and google.com were frequently visited in a prior study of the browsing habits of blind web users [8], and mybus.org is a popular site for checking the bus in Seattle, where most of our evaluators live. The authors feel that these tasks are representative of those that a screen reader user who is away from their primary computer may want to perform.

We did not test the system with blind individuals who would like to learn to use a screen reader but cannot afford one. Using a screen reader efficiently requires practice and our expectation is that if current screen reader users can use WebAnywhere then others could likely learn to use it as well.

6.0.1 Task 1: Restaurant Phone Number on Google

Participants were asked to find the phone number of the Volterra Italian Restaurant in Seattle by using google.com. Participants were told to search for the phrase “Volterra Seattle.” The phone number of the restaurant can be found

on the Google results page, although some participants did not notice this and, instead, found the number on the restaurant’s home page. This task represented an example of users wanting to quickly find information on-the-go.

6.0.2 Task 2: Gmail

Participants checked a web-based email account and located and read a message with the subject “Important Information.” Participants first navigated to the gmail.com homepage and entered a provided username and password into the appropriate fields. They next found the message and then read its text. This task involved navigating the complex pages of gmail.com that include a lot of information and large tables.

6.0.3 Task 3: Bus Schedule

Participants found when the 48 bus would next be arriving at the intersection of 15th Ave and 65th St using the real-time bus tracking web page mybus.org. Participants first navigated to the mybus.org homepage, entered the bus number into a text input field and clicked the submit button. This brought them to a page with a large list of links consisting of all of the stops where information is available for the bus. Participants needed to find the correct stop among these links and then navigate to its results. This task also included navigating through large tables of information.

6.0.4 Task 4: WebAnywhere Survey

The final task asked participants to complete a survey about their experiences using WebAnywhere. They completed the survey about WebAnywhere using the WebAnywhere screen reader itself. This task involved completing a web-based survey that consisted of eleven statements and associated selection boxes that allowed them to specify to what extent they agreed or disagree with the statement on a Likert scale. The survey also included a text area for additional, free-form comments. For this task, the researchers left the room, and so participants completed the task completely independently.

6.1 Study Results

All participants were able to complete the four tasks. Most users were not familiar with the pages used in the study and found it tedious to find the information of interest without already knowing the structure of the page. However, most noted that this experience was not unlike using their usual screen reader to access a page which they had not accessed before. Some participants noted functionality available in their current screen reader that would have helped them complete the tasks more quickly. For example, the JAWS screen reader has a function for skipping to the main content area of a page.

Participants who were already familiar with the web pages used to complete the tasks in the study, were, unsurprisingly, faster at completing those tasks. For instance, several participants were frequent GMail users and leveraged their knowledge of the page’s structure to quickly jump to the inbox. In that example, skipping through the page using a combination of the next heading and next input element shortcut keys is an efficient way to reach the messages in the inbox.

Figure 6 summarizes the results of the survey participants took as part of task 4. Participants all felt that WebAny-

where was a bit tedious to use, although many mentioned in a post-study interview that it was only slightly more tedious than the screen reader to which they are accustomed. Most agreed that mobile technology for accessing the web is expensive and most find themselves in situations where a computer is available but they cannot access it because a screen reader is not installed on it. The main exception was a skilled computer user who is involved in the development of the NVDA screen reader² and carries a version of this screen reader with him on a USB key. He was uniformly negative about WebAnywhere because he thought its current form provided an inferior experience relative to NVDA and because the mobile version of NVDA already works for him on most computers that he has tried. Most of our participants could see themselves using the system when it is released.

6.2 Discussion

Participants felt that WebAnywhere would be adequate for accessing the web, although none were prepared to give up their normal screen reader to use it instead. This was the expected and desired result. One participant remarked that the system would be useful for providing access when he is visiting a relative where he would not be comfortable installing new software. Another participant who was testing a preliminary version of a web-based version of Serotek’s SA-to-Go [29] said that he could see himself using either system for mobile access and would base his decision on the price of the Serotek tool. After this study session, the Serotek system has been made freely available.

Several participants completed the study after the Serotek tool had been released for free. They said that they could see themselves using both tools depending on their situation. For instance, if they only needed to check the web to retrieve a phone number or an email, they would probably use WebAnywhere because it does not take as long to load. They would also use WebAnywhere if they were on a machine on which SA-to-Go would not run. The Serotek tool would be particularly useful if they needed to access applications other than the web.

Participants did not generally mention the latency of the system overall as a concern, but some were confused when a sound or web page took a while to load because, during that period, WebAnywhere was silent. We can address this by having the system periodically say “content loading, X%” while content is loading. One participant mentioned that the latency of speech repeated to him while typing was bothersome. We have improved this by prefetching the speech representation of letters and punctuation that result when users type.

Others noted that errors in the current implementation occasionally produced incorrect effects and that WebAnywhere lacks some of the functionality of existing screen readers. These shortcomings can be addressed in future versions of WebAnywhere. The latency of the system can be improved through more aggressive prefetching, smarter caching and targeted compression of sounds. Current screen readers have spent great effort in providing a large array of shortcuts and other features to improve the experience of browsing the web with a screen reader. With time, WebAnywhere can implement these as well.

²The participant was recruited from an online list without prior knowledge of this relevant experience.

Most importantly, participants successfully accessed the web using WebAnywhere; future versions will seek to further improve the user experience and functionality.

7. SYSTEM AVAILABILITY

The complete code for WebAnywhere has been released as an open source project on Google Code licensed under the minimally-restrictive New BSD License. It is accessible at the following URL:

<http://webanywhere.googlecode.com>

We have begun an initial test of an installation of the system as a service that is hosted by the University of Washington. Thus far it has been made available to a limited number of users for testing purposes, but we will make it available to everyone in May 2008.

8. FUTURE WORK

As we continue the development of WebAnywhere, we will seek to improve the user experience that it provides. The participants in our user evaluations suggested that we add more of the features offered by other screen readers. Because WebAnywhere is web-based we can quickly introduce new features and iterate on its design. We will also continue exploring methods for supporting additional content types that are usually handled by external applications. Users universally requested that the accessibility of web content be improved, and so we will add support for accessibility improvement using the Accessmonkey Framework [7]. As we continue to improve WebAnywhere we will do so with close consultation with blind web users.

We also plan to release WebAnywhere so that all blind users and web developers can use it access the web from computers that are not their own and review content that they create to make it more accessible. The immediate improvements that we are making to the system include improved caching and prefetching, and security improvements designed to reproduce in WebAnywhere the security policies of the host browser [9]. Ensuring that WebAnywhere can be efficiently hosted as a web service is important for its successful deployment. We are currently completing studies designed to test the performance of WebAnywhere and the contribution of various caching and prefetching techniques. The results of these studies will help determine the resources required to release the system.

Finally, we plan to ensure that WebAnywhere can operate on cellphones and other mobile devices with built-in web access. Most of these (often proprietary) devices do not have a screen reader available for them, but many support web access. Most mobile devices today use browsers with limited functionality that cannot play sound from within the browser, but this is beginning to change. As more powerful mobile devices become popular, WebAnywhere may prove to be a quick way to enable accessible browsing on these new platforms before fully-functional screen readers are developed for them.

9. CONCLUSION

The WebAnywhere web-based, self-voicing web browser enables blind individuals otherwise unable to afford a screen reader and blind individuals on-the-go to access the web

from any computer that happens to be available. WebAnywhere is able to be run on most systems, even public terminals on which users are given few permissions. It's small startup size means that users will be able to quickly begin browsing the web, even on relatively slow connections. Participants that evaluated WebAnywhere were able to complete tasks representative of those that users may want to complete on-the-go.

10. ACKNOWLEDGEMENTS

This work has been supported by National Science Foundation Grant IIS-0415273 and a Boeing Professorship. We thank T.V. Raman, Jacob O. Wobbrock, Sangyun Hahn, Lindsay Yazzolino, Anna C. Cavender and Don Swaney for their comments. We thank Shaun Kane and Karl Koscher for lending us equipment. Finally, we thank our study participants for their invaluable feedback.

11. REFERENCES

- [1] Adobe shockwave and flash players: Adoption statistics. Adobe (June 2007). http://www.adobe.com/products/player_census/.
- [2] Air Foundation (February 2008). <http://www.accessibilityisright.org>.
- [3] Arif, A. Phproxy (2007). <http://whitefyre.com/poxy/>.
- [4] Atterer, R., Wnuk, M., and Schmidt, A. Knowing the user's every move - user activity tracking for website usability evaluation and implicit interaction. In *Proceedings of the 15th International Conference on the World Wide Web (WWW '06)*. Edinburgh, Scotland, 2006, 203–212.
- [5] Braille Sense. GW Micro (2007). http://www.gwmicro.com/Braille_Sense/.
- [6] Bertot, J. C., McClure, C. R., Jaeger, P. T., and Ryan, J. *Public Libraries and the Internet 2006: Study Results and Findings*. Technical report, Information Use Management and Policy Institute, Florida State University (September 2006).
- [7] Bigham, J. and Ladner, R. Accessmonkey: A collaborative scripting framework for web users and developers. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (W4A '07)*. 2007, 25–34.
- [8] Bigham, J. P., Cavender, A. C., Brudvik, J. T., Wobbrock, J. O., and Ladner, R. E. WebinSitu: A Comparative Analysis of Blind and Sighted Browsing Behavior. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*. Tempe, Arizona, USA, 2007, 51–58.
- [9] Bigham, J. P., Prince, C. M., and Ladner, R. E. Engineering a self-voicing,web-browsing web application supporting accessibility anywhere. 2008. Available at: <http://webinsight.cs.washington.edu/publications/webanywhere-engineering.pdf>. *Submitted*.
- [10] Bigham, J. P. and Prince, C. M. WebAnywhere: A Screen Reader On-the-Go In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility (Demonstration)*. Tempe, Arizona, USA, 2007, 225–226.
- [11] Chen, C. Fire vox: A screen reader firefox extension (2006). <http://firevox.cleworld.net/>.

- [12] Coyne, K. P. and Nielsen, J. Beyond alt text: Making the web easy to use for users with disabilities (2001).
- [13] Email2me. Across Communications (2007). <http://www.email2phone.net/>.
- [14] GOOG 411. Google Labs (2008). <http://labs.google.com/goog411/>.
- [15] Google Analytics. Google, Inc. (2008). <http://analytics.google.com/>.
- [16] Google Documents. Google, Inc. (2008). <http://docs.google.com/>.
- [17] JAWS 8.0 for Windows. Freedom Scientific (2006). <http://www.freedomscientific.com>.
- [18] Linux Screen Reader (LSR) (2006). <http://live.gnome.org/LSR>.
- [19] Mankoff, J., Fait, H., and Tran, T. Is your web page accessible?: a comparative study of methods for assessing web page accessibility for the blind. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '05)*. Portland, Oregon, USA, 2005, 41–50.
- [20] Miyashita, H., Sato, D., Takagi, H. and Asakawa, C. Aibrowser for multimedia: introducing multimedia content accessibility for visually impaired users In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '07)*. Tempe, Arizona, USA, 2007, 91–98.
- [21] Mobile Speak Pocket. Code Factory (2007). http://www.codefactory.es/mobile_speak_pocket/.
- [22] National association for the blind, India (2007). <http://www.nabindia.com/>.
- [23] NVDA screen reader. NV Access Inc. (2007). <http://www.nvda-project.org/>.
- [24] Orca Screen Reader: the GNOME Project (2008). <http://live.gnome.org/Orca>
- [25] Petrie, H., Hamilton, F., King, N., and Pavan, P. Remote usability evaluations with disabled people. In *Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06)*. Montreal, Quebec, Canada, 2006, 1133–1141.
- [26] Ramakrishnan, I.V., Stent, A., and Yang, G. Hearsay: Enabling audio browsing on hypertext content. In *Proceedings of the 13th International Conference on the World Wide Web (WWW '04)*. 2004.
- [27] Schiller, S. Sound manager 2 (2007). <http://www.schillmania.com/projects/soundmanager2/>.
- [28] Scribd (2007). <http://www.scribd.com/>.
- [29] Serotek system access mobile. Serotek (2007). <http://www.serotek.com/>.
- [30] Talklets. Hidden Differences Group. <http://www.talklets.com/>.
- [31] Taylor, P. A., Black, A. W., and Caley, R. J. The architecture of the the festival speech synthesis system. In *Proceedings of the 3rd International Workshop on Speech Synthesis*. Sydney, Australia, 1998.
- [32] Thylefors, B., Negrel, A., Pararajasegaram, R., and Dadzie, K. Global data on blindness. *Bull. World Health Organ.*, 73, 1 (1995), 115–121.
- [33] VoiceOver: Macintosh OS X (2007). <http://www.apple.com/accessibility/voiceover/>.
- [34] Webaim screen reader simulation. WebAIM (2007). <http://www.webaim.org/simulations/screenreader>.
- [35] Window-Eyes. GW Micro (2006). <http://www.gwmicro.com/Window-Eyes/>.
- [36] Windows Narrator: Microsoft’s Windows XP and Vista (2008). <http://www.microsoft.com/enable/training/windowsxp/narratorturnon.aspx>.

APPENDIX

A. USER EVALUATION RESULTS

Question	Participants							
	A	B	C	D	E	F	G	H
1	4	2	3	3	3	5	3	3
2	2	3	5	1	1	5	3	2
3	5	4	3	4	4	1	3	1
4	4	4	4	5	4	1	4	5
5	3	5	2	2	5	1	4	5
6	5	2	2	4	4	1	4	3
7	5	1	5	5	5	1	5	5
8	4	3	3	4	1	5	2	4
9	5	2	5	5	5	1	4	5
10	5	4	3	4	5	1	3	5
11	5	4	3	3	5	1	2	5

Table 1: Full results of the survey given to user evaluation participants indicating their responses on a Likert Scale from 1 (strongly disagree) to 5 (strongly agree). Question numbers refer to Figure 6.