

Analysis and Improvement System WebAnywhere

Carla Fernanda da S. Sampaio¹ and Luís César Martini²

¹*School of Electrical and Computer Engineering, State University of Campinas, Av. Albert Einstein, Campinas 400, Brazil*

²*Department of Communications, State University of Campinas, Av. James Maxwell 30, Campinas, Brazil*

Keywords: Accessibility, WebAnywhere, Blind Users, Web, Assistive Technology.

Abstract: This work presents the WebAnywhere system which is a screen reader for specific website for people with visual impairments. This research aims to enhance this program aimed at developing new features and functionality for the user to be able to perform blind download web files such as images and videos through the reader WebAnywhere. This project also involves testing to facilitate the use of this tool and adaptive display a consistent methodology to the context of this study.

1 INTRODUCTION

The Internet offers people unprecedented access to information and data, namely access and use the Web has become indispensable for participation in society.

People who do not have disabilities have very easily to access web content, just use the mouse to get the elements of interest and ignore or read the information or announcements relevant to a site like flipping through titles, search or select files and images but the question arises of how the visually impaired could ever hope to use the web (Lévy,1999).

Blind people need to use software programs called screen readers that allow access control and blind people in a computer system, the information that is normally transmitted on the screen will be presented to the blind via voice synthesis provided access and greater degree of independence in social life such technology expands the universe of deficient affective life with their interpersonal relationships, as well as in academia, and in his work (Shawn, 2006).

But due to lack of accessibility and standardization of the content of webpages screen readers are unable to guarantee a fair presentation of such information as well as access to data that would be relevant to those users, most sites still have several barriers that hinder accessibility, and most suitable for adaptive web technologies are not sufficiently accessible to people with disabilities, making it difficult or impossible for them to use for

these new software and applications that have emerged in the market are proprietary products that do not have standardized according features the recommendations of the World Wide Web Consortium (Hersh and Johnson, 2005).

The technology Adobe Flash Player in a general sense can be one of the most used applications and available on the web and has achieved ubiquity in Internet content emerging as a plug-in that allows various types of multimedia in the browser.

However the use of non-standard format caused significant accessibility problems for some people, because despite the prevalence of open formats on the web and the existence of an open specification to SWF format Flash Player (Regan and Kirpatrick, 2006).

The content developed in Flash is completely inaccessible to assistive technologies that aim to convey information to users with disabilities in away that is different from the content developer offered.

And the source code for the Flash Player is not available to the public, because there are no alternatives to the full-featured players, facilities and implementation of assistive technology for working with Flash is very limited.

Another problem facing the visually impaired is the reading and understanding of graphics, there are many kinds of images on the Internet, also including animated GIF or spacers which serve to limit the elongation of a cell which is purely decorative and not have any relevant information (Hersh and Johnson, 2008).

In this case we use the alt attribute and it needs to be located within the attribute tag so that a

screen reader can identify alternate text, however blind users cannot capture the graphics for your computer as people have no such limitations (Radabaugh,1993).

So the idea of this project is to improve the WebAnywhere screen reader for blind users are able to download videos from the YouTube site and catching graphics.

The software WebAnywhere screen reader is a web based that can be used by visually impaired people to access the Internet from virtually any computer with a web connection and audio output, designed to ensure more freedom for users to access content that want anywhere and cover a wider digital inclusion because it is an open source project and unlike software that must be installed on your computer, this is an online application that makes browsing available on virtually any computer and any operating system. This paper will report how the operation of a screen reader for specific website, as well as analysing some concepts of DOM API that is essential to the project and finally analyze the system WebAnywhere proposing implementation of new features.

2 OPERATION SCREEN READER SPECIFIC TO WEB

Assistive technologies are important instruments for evaluation and necessary for the web developer.

Screen readers are tools through which primary care throughout the site content is channeled. Blind people use screen readers to obtain information that is displayed on the computer screen from speech synthesis (Cook and Hussey, 1999).

However these software are hard to use because they must be able to read any application or resource, for pages with high level of complexity of creating demand that is used certain keyboard commands that can become difficult for a blind user to use.

Formerly screen readers used the technology off-screen to view web content, this method determines the attributes of a pixel, which includes the value of a pixel display characteristics (bold, italic,underline).

This type of player is able to capture the information on a computer screen via the video driver and display to the user, which is nothing more than a database of all information displayed on the screen.

However with the passage of time it was realized that this technology did not meet the actual expectations of blind users because the data were not

transmitted by default (Regan and Kirpatrick, 2006). Starting found that the need for screen readers being able to work with the DOM API that is created by the World Wide Web Consortium, so instead of reading the screen, the screen reader will use the DOM to map the elements to HTML display information to the user, presenting the data in twodimensional form, converting the information flow for a dimensional character (Weste, 2009).

All information will be presented to users through a voice synthesizer and this process called linearization of the page where the screen reader realizes the capture of data in the HTML page ignoring all its brands, seeking only the text values of some attributes as alt and title of the document, which is the result of a text file, it is considered the linearized version of the page.

2.1 API DOM

The Document Object Model or DOM is considered a multi-platform representing all markups in HTML, XHTML and XML.

Once indexed these markings become elements of a tree that can be manipulated via API which is what happens when you use scripts or programs to change functionalities of a page: content, structure or style sheet (Keith, 2002).

Every modern browser contains a DOM enabling JavaScript programs can read and change the web page the user is currently viewing, with a structured approach to object-oriented elements of the page content and providing methods to add and remove such objects, that allows you to create dynamic content.

It is designed to be independent of any programming language, and this is very important due to the fact several programming languages use to perform the access on a given page. Figure 1 shows the structure of the DOM architecture representing its markings indexed HTML elements of a tree that can be manipulated via API, with the linearization of the markings so that it can be fitted initially by a browser.

Every DOM tree has a class called node.h, which has several categories to ensure the rendering code in the browser that we are the documents, and text elements.

- Document: It has three different classes, used by all XML and HTML documents and is the most important node of the DOM.
- Elements: the markings are provided in HTML that will be represented in the tree.
- Text: The text displayed in the elements

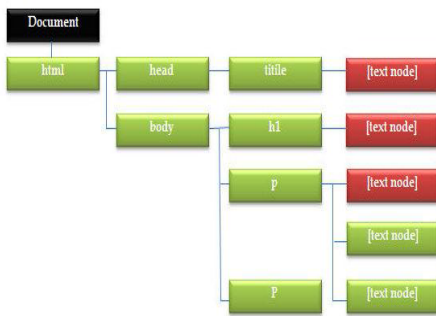


Figure 1: Representation of an HTML document and itsDOM tree.

This architecture is not the layout that the user sees in the browser. This is the basis for another tree which is what browsers display screen called tree rendering.

2.2 The Tree DOM

One of the features of the DOM is that it represents its entirety as if I was referencing items of a family tree. Where has a single root node and all nodes in this tree, except the root node has a single parent.

Each node has a list of child nodes, the representation of the tree describes the relationships between family members and uses conventions, as a father, son, brother.

These can be used to represent some fairly complex relationships, providing all the functionality that is needed to modify, remove or create new content into the pages. It is very important for access and retrieval of relevant information structure and content of a document, because all the information reflected in this API document data (Keith, 2002).

The DOM is a two-way street, because you can see the contents of a document and simultaneously update the same (Sampaio, 2012). For the screen reader can access better navigate the website he needs to understand the exact structure of the tree to access and manipulate its elements, to better understand the navigation will be used as the reference following HTML code:

```
[1]<htmlxmlns="http://www.w3.org/1999/xhtml">
[2] <head>
[4] <title>Core API<</title>
[5] <script src="js/file.js">
[6] </head>
[7] <body>
[8] <h2>Tree</h2>
[9] <div>
[10] <ul>
```

```
[11] <li id="item 1">
[12] Primeiro
[13] <span style ="color:blue;">item</span>
[14] <ul>
[16] <li id="Item 1.1">Item 1.1</li>
[17] <li id="Item 1.1">Item 1.2</li>
[18] <li id= "Item 1.1">Item 1.3
[19] <ul>
[20] <li id="Item 1.3.1">Item 1.3.1
[21] <ul>
[22] <li>
[23] </ul>
[24]</li>
[25] <li id="Item 2">Item 2</li>
[26] <li id="Item 3">Item 3</li>
[27] </ul>
[28] </body>
[29] </html>
```

The document element is presented within an html element, which is the parent body and head and show html as its parent (Weste, 2009). And so on down the hierarchy of the document, with each element pointing to his direct descendants as children, and his direct ancestor as a parent:

- The title tag and script are represented as children of head.
- The body has as children h2 and div tags.
- The h2 tag is the parent node tree
- Div tag is the parent of ul.

This HTML is presented visually to something in figure 2:

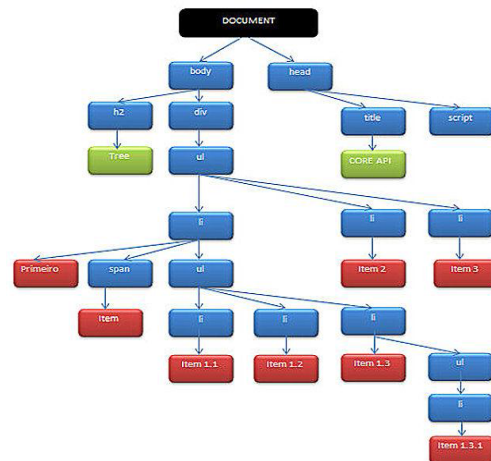


Figure 2: Representation of an HTML document and its DOM tree.

The screen reader must navigate through the page and map all the tags and textual data, this process can be called digitalization, is a linear, leftto- right,

considering the chain as a two-dimensional object (Branjar, 2012).

The task is to add another dimension in time, thus expressing different levels of the tree.

3 WebAnywhere SYSTEM

The screen reader must navigate through the page and map all the tags and textual data, this process can be called digitalization, is a linear, left-to-right considering the chain as a two-dimensional object.

The task is to add another dimension in time, thus expressing different levels of the tree. This application was launched in 2008 and aims to provide Internet browsing on any computer because their operation occurs directly through the browser. Thus the user does not need to install any software on the client machine.

The application can be run on any operating system and any device that is enabled for web, which includes devices, the requirements for their use is that the computer or device are connected to the Internet and have audio out. In the market there are many softwares that allow web access for blind, yet few are those who have good functionality and consequently a low cost (Bigham et al., 2008).

Another drawback is that the blind user cannot access the web anywhere, such as screen readers are not or cannot be installed on any terminal. But with the WebAnywhere users can perform everyday tasks, such as accessing e-mail, check the starting time of a transfer of a plane, find the phone number of a store, and access to the web anywhere.

3.1 Analysis and Operation System WebAnywhere

When the user accesses the main page of WebAnywhere content is automatically loaded, displaying the speech of welcome. The system runs as a web application that requires minimal permissions, which differs from the WebAnywhere traditional screen readers is the fact that he can directly interact with the DOM.

It has the ability to reach each element of the DOM of a page loaded, and it uses search algorithms in depth first order, this mechanism is run from top to bottom and from left to right of the page displayed.

The system does not access the browser interface, however it has its own address bar displays the URL where the user accesses.

As the main adjective WebAnywhere system is run on any computer that has Internet access and audio output, developers bother with an interface easy to use, considering its use in public systems where users have minimum permissions. Its interface is an independent web browser that runs inside a regular web browser (figure 3):



Figure 3: WebAnywhere is a standalone browser within a browser (Bigham et al., 2008).

The system consists of three components: (i) on the client side the JavaScript language that the user will interact with and support the operation of sound and its reproduction, (ii) on the server side the text is converted to speech and caching, and (iii) proxy server side there is a transformation of web pages for a local server to prevent violations of the security policy of the same origin imposed by most web browsers (Bigham et al., 2008).

The charging time of the data to the main is approximately 100 Kb, representing less than 5 seconds. It is then the use of independent browser that works inside a web browser. The execution takes place at the server, which has the function to convert text into speech for user content and bring all of the proxy server, client-side interface is displayed to the user, along with playback of sounds.

Its development was conducted in the language cross-browser JavaScript that is downloaded from the server, which causes it to run on most modern browsers, interacting directly with the DOM and capturing all key events, keeping track of the browser window, but for this to occur the pages are loaded PHP Proxy link. Its operation below in figure 4:

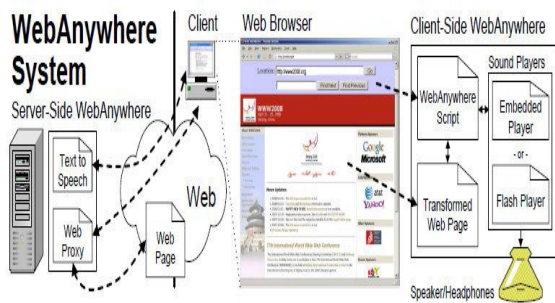


Figure 4: Model for the basic functioning of the WebAnywhere (Bigham et al., 2008).

Turn sounds are played using Adobe Flash via Sound Manager this has an API that determines when speech should begin and end. The API performs the interaction between JavaScript and Flash code, but all discourse is produced primarily in the server using the speech synthesis system Festival which converts all files to MP3 format.

This extension is compatible with most audio players for browsers, and create small files that help low latency and sounds are stored in cache server so there is no need to be produced when requested again. The system does not access windows that are not part of its focus display, but the focus can often be displaced due to pop-up, or even dialog boxes that appear during navigation.

When this occurs, the system will guide the user back to the tool when it is lost, the user can press escape and then ALT-TAB back to the system with the focus again.

3.2 Specifications on the WebAnywhere

WebAnywhere the system has the ability to skip titles, graphics guides, or even read the page from top to bottom, edit input fields, report the amount of images on a page and the name of the same images if they have text equivalents (Bigham et al., 2008).

One limitation of WebAnywhere is he like most of the existing screen readers cannot work on sites that have Flash elements, because it's own boundaries that Flash has, so if the user accesses a page that has Flash on the property program will skip this part if the page has form fields like group selection system has the ability to read the name of group selection, but the user will not have the ability to choose an option from the keyboard.

These specifications were described using as basis the beta version of the system, however errors may occur during the use of the same, in which case

you should use the alpha version. Regarding the interaction between the user and the web pages, WebAnywhere has several features for improved accessibility through the use of the keyboard using the following set of commands:

CTRL-L: Move the cursor to the box address (URL) where the user enters the desired site.

Down Arrow: Reads the next element of the page.

Up Arrow: Reads the previous element of the page.

TAB: Skips to the next page link.

CTRL-H: Skip to the next header.

CTRL-I: Skips to the next input element.

CTRL-R: Skip to the next row by cell when in a table.

CTRL-D: Jump to the next column when a cell in a table.

Page Down: Continuously reads from the current position.

Home: Reads continuously from the beginning of the page.

CTRL: Silence WebAnywhere and pause the program.

4 DESIGN METHODOLOGIES

Initially was studied and analyzed the architecture and logic of the system WebAnywhere alpha version after this test and adjustments were made in the source code.

Aiming to achieve better usability and a positive evaluation by the users as well as a good degree of satisfaction, ease of use, new ideas, a survey of requirements with some blind users that the system meets your needs, and define ways in which it should operate.

Because it is a specific system for Internet, it was necessary to study various functions of JavaScript and DOM API, that allowed scripts could be set to update the content, structure, and document style mechanisms search and filter and display a consistent methodology to the context of this study.

Besides analyzing the main issues of interface, while this is not technically the problem of the developer or designer, is a detail necessary to understand the practical obstacles that stand in the way of graphics and videos accessible.

A blind or visually impaired definitely has difficulty given that it is normal for these elements because they are dealing with layers of abstraction:

- The computer hardware. (With screen readers, large keys and unusual are the norm),
- The operating system of the machine,
- The application software such as a browser or reader, possibly with multiple windows open,

- Accessibility features content.

Thus a first package of functionality has been established in which blind users can capture graphic elements and perform download videos from the Internet using the system for such WebAnywhere.

The development was started in parallel and further tests were made during the project, through programming and architecture JavaScript DOM scripts were programmed to update the content, structure, style research papers and filtering mechanisms to display a methodology consistent with the context of this study.

4.1 Videos in Flash Format

When it refers to access web content with flash, obstacles for users with disabilities have two sources: issues associated with design issues and assistive technologies (Regan and Kirpatrick, 2006).

Assistive technologies are often only adapted to more general web technologies after they are common enough to generate demand for relevant assistive technology. Designers have a unique vision to create, imagine, conceptualize, and translate the information into visual layout and graphics, however they have an immense difficulty to convey that experience to the disabled in a way that does not harm, and ensuring accessibility to this information understandable public.

One way out of the blind can enjoy this content is through the download when the user access to the YouTube page, the video URL accessed will be automatically copied to the new bar in the URL field, so the user will only need to press the download button here, running on a simple interface, fast and enjoyable for blind users being fed by speech synthesizer Festival which will transmit the information to the user. The videos are saved in the user's computer to a local folder in Flash format, which can be executed by any player on the user's computer. The download time will depend on the connection that you are using to get access to videos of the YouTube page was also a need to study the YouTube API. In Figure 5 it is possible to see the creation of a new address bar system:

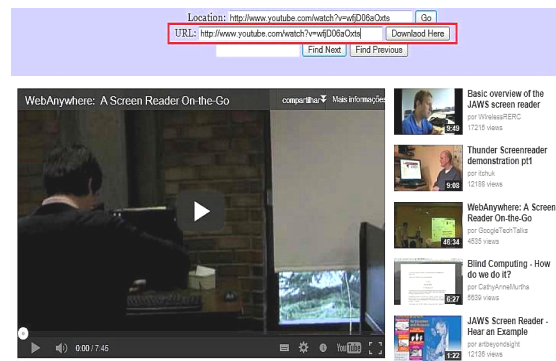


Figure 5: New address bar system WebAnywhere with the URL of the desired video.

4.2 Graphics Elements

Users who have low vision cannot perform the capture of the graphics by assistive Technologies (Sonza, 2008), so we built a new command for which the system was able to track the graphics architecture using the DOM and JavaScript programming, this way the user can perform download when the keyboard command CTRL C fires, and thus save the image by opening a dialog box.

However, it is necessary for the user to press the ENTER key to finalize the process of capturing the image is saved on the desktop of the user's computer the system will not lose focus, because it is a very quick process, after downloading the system will continue to read the web page from the current position.

All this is powered by voice synthesizer that will pass the information to the user. In Figure 6 the new functionality within the WebAnywhere system.



Figure 6: Image mapped and being activated by CTRL-C keyboard and opening the dialog (Sampaio, 2012).

This research helped to understand the needs of blind users for better usability of the resources offered by the Internet and showed the importance of having a web development more accessible because it makes the information available to the blind through the use of screen readers.

The web is an indispensable tool in today, enabling disabled people are able to use and contribute web content is an important aspect of making it accessible. In the following work were developed initially specified in the commands that allow blind users are able to make downloading pictures and videos, however these features are still in the experimental stage to give disabled better usability, interactivity, and especially in order to ensure greater access to a wider range of resources and ensure the visually impaired the same capabilities to interact with the Web, as well as people who do not have such limitations.

5 CONCLUSIONS

Through these studies were developed in the new commands that allow blind users are able to perform downloading and printing files as pages, images and videos.

That gave the blind a better use of resources the internet offers. As for the new features implemented, several improvements must be made in order to enable better usability and interactivity, mainly in order to ensure greater access to a wider range of resources.

Another important point is the need to use existing resources in creating accessibility of sites. For without them, however there are specific features in the browser, some elements remain inaccessible.

Finally, this research demonstrates how important it is to provide the same capabilities visually impaired to interact with the Web, as well as people who do not have such limitations. This brought to blind users an autonomy and a better use of resources the Internet offers.

ACKNOWLEDGMENTS

This work was supported by CAPES Foundation.

REFERENCES

- Bigham, J. P., Prince, C. M., Lander, R. E. *Addressing Performance and Security in a Screen Reading Web Application That Enables Accessibility Anywhere. In Proc. of the Eighth International Conference on Web Engineering* p. 273-284, 2008.
- Bigham, J. P., Prince, C. M., Lander, R. E. *Webanywhere: A Screen Reader On-the-Go. In Proc. of the Intl. Cross- Disciplinary Conf. on Web Accesibility (W4A)*, 2008.
- Branjar. 2012. *Introduction to DOM* Available: <http://www.brainjar.com/dhtml/intro/>.
- Cook, A. M.; Hussey, S. M., *Assistive Technologies: Principles and Practices*. St. Louis, Missouri. Mosby Year Book, Inc. (1995).
- Hersh, M. A., Johnson, M. A., *Assistive Technology for Visually Impaired and Blind People*. 1. ed USA: Springer, 2008. 50-70p.
- Hersh, M. A., Johnson, M. A. *Information technology, accessibility and deaf blind people, proceedings of Association for Advancement of Assistiv Technology in Europe Annual Conference, Lille, France(2005)*.
- Keith, J., *DOM Scripting: Web Design com JavaScript e Document Object* . 1.ed. USA: friends of ed, 2002. 25p.
- Lévy, P., Ciberultura. São Paulo: Ed. 34, 1999.
- Radabaugh, M. P., *Study on the Financing of Assistive Technology Devices of Services for Individuals with Disabilities - A report to the president and the congress of the United State, National Council on Disability*, March, 1993.
- Regan, B.; Kirpatrick, A. *Web Accessibility Web Standards and Regulatory Compliance: Web Accessibility: Web Standards and Regulatory Compliance*. 1d. USA: 2006. 2.p.
- Sampaio, S. F. C., *Artigo Aperfeiçoamento do Sistema WebAnywhere: um leitor de tela para deficientes visuais específico para web. Simpósio de Inovação Tecnológica na Educação*, Brasil (2012).
- Shawn, L. H., *Web Accessibility - Web Standards and Regulatory Compliance: 1d. USA: 2006. 2.p.*
- Sonza, A. P. *Ambientes virtuais acessíveis sob a perspectiva de usuários com limitação visual. 2008. Tese (Doutorado em Informática na Educação) – Faculdade de Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.*
- Weste, M., *Traversing the DOM Article*, n.46, p.1-3, Febr 2009. Available: <http://dev.opera.com/articles/view/traversing-the-dom/>