Human Computer Interaction for Visually Impaired Users in Web Based Applications

A proposed solution to make rich internet application completely accessible to no vision/low vision users.

Yogesh Lele, Chirag Kariya, Nupur Kale, Aarti Nandgaonkar, Prof. Madhura Phatak

Abstract—This paper focuses on making any web based system Human computer Interaction (HCI) compliant for visually impaired users thereby achieving a point of conjunction between man and machine. Currently not many web-based apps have means to provide accessibility. Our system aims to make existing web based systems as well as systems to be developed, accessible by using screen reader as an assistive technology and keyboard navigation. We propose to incorporate WAI-ARIA standards into web based system which acts as an accessibility API. It intercepts user inputs and dynamic web contents and generates information to screen reader which outputs it as voice.

Index Terms— Accessibility, Blind User, Human Computer Interaction, Key-board Navigation, On-Screen Reader, WAI-ARIA.

1 Introduction

With computer and internet being everyday part of workplaces almost everywhere, it becomes difficult for low vision/no vision users to fit in the working environment. Web based systems which are platform independent, are generally rich in User Interface but they lack accessibility. Normal people can understand it because of their cognitive senses. However blind people have no such accessibility mechanisms that can bring them par with the normal users. The means by which humans interact with computers continues to evolve rapidly. The making of such systems focuses on making web contents, human computer interaction compliant, so that even visually impaired people can use normal systems without addition of special hardware. To achieve this point of conjunction between man and machine we propose to make use of screen reader as an assistive technology and an accessibility API called WAI-ARIA [1] which can be incorporated in existing as well as new web based systems without changing the semantics of current system.

2 Related Work

Following table shows HCI studies about modalities [3] of human perception with pros and cons.

TABLE 1
HUMAN PERCEPTION FEEDBACK MODALITIES

	Pros	Cons
Visual	Parallel in space; large information transfer	Active eye contact necessary
Acoustic	Enforces attention allocation; enables the perception of background activities; important for visually impaired people	Noise through environment; linear in time; exists only for a short time span
Haptic	Object recognition; texture and surface perception	Contact with objects necessary

2.1 Assistive Technologies

After Braille, no invention has enabled blind and visually impaired people to communicate as effectively as the assistive technologies that make computers and the Internet accessible. Among these are:

2.1.1 Screen Readers

Screen readers speak aloud, what is on a computer screen, including desktop icon labels, document contents, and drop-down and tool bar menu items. They also speak each keystroke; provide auditory cues and an audible hierarchy for navigating within and among applications. However their drawback is that they don't react intelligently to the dynamic contents on the screen.

2.1.2 Braille Translation Software

Braille translation software converts electronic files into braille that can be read on a refreshable display or printed on a braille embosser. Blind computer users use it to read text output. This solution however, requires involvement of an extra hardware.

2.2 WAI-ARIA

WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications)

As the name suggests, WAI ARIA is an accessibility API provided by W3C and it defines a way to make Web content and Web applications more accessible to people with disabilities. It especially helps with dynamic content and advanced user interface controls developed with Ajax, HTML, JavaScript, and related technologies. Currently certain functionality used in Web sites is not available to some users with disabilities, especially people who rely on on-screen readers and people who cannot use a mouse. More specifically, WAI-ARIA provides a framework for adding attributes to identify features for user interaction, how they relate to each other, and their current state. WAI-ARIA describes new navigation techniques to mark regions and common Web structures as menus, primary content, secondary content, banner information, and other types of Web structures. For example, with WAI-ARIA, developers can identify regions of pages and enable keyboard users to easily move among regions, rather than having to press Tab many times. [1]

WAI-ARIA provides Web authors with the following: [2]

Roles to describe the type of widget presented, such as "menu," "tree tem," "slider," and "progress meter".

Roles to describe the structure of the Web page, such as headings, regions, and tables (grids).

Properties to describe the state widgets are in, such as "checked" for a check box, or "has popup" for a menu.

Properties to define live regions of a page that are likely to get updates (such as stock quotes), as well as an interruption policy for those updates—for example, critical updates may be presented in an alert dialog box, and incidental updates occur within the page.

A way to provide keyboard navigation for the Web objects and events, such as those mentioned above.

3 Proposed Work

Every existing web based system can be made accessible for a blind user by this proposed solution instead of creating new interface from scratch. A web page is nothing but a DOM Tree structure with its nodes as a HTML Tags. ARIA can be thought of as a meta-markup language that is incorporated in these tags. Every element can be assigned ARIA Role, State and Property [1]. Doing this doesn't change the semantics of the DOM structure [4] but it provides the Screen reader with additional information, by means of which, while reading it would add up appropriate messages. For example, Consider a HTML textbox created using <input> tag. When read by screen reader, it acknowledges the presence of text box and reads its contents. Now given the pretext that the contents inside it are going to change dynamically, the screen reader is not going to understand when they actually change. As mentioned previously imparting ARIA roles such as 'live region' solves this difficulty. By doing this ARIA API throws an interrupt to screen reader whenever the contents of the text box changes. This interrupt is served by the Screen reader by temporarily suspending the current text being read and reading this dynamically changed text. In absence of ARIA role this would not have been possible. The following schematic diagram adds up onto our thesis.

Incorporating WAI-ARIA would ultimately help the developers to program the on-screen reader and control what it reads. This would enable any blind user to work on the web application with ease. The on-screen reader would direct the user to perform appropriate actions with the help of proper acknowledgements. Web applications being run in the browser make them Operating system independent.

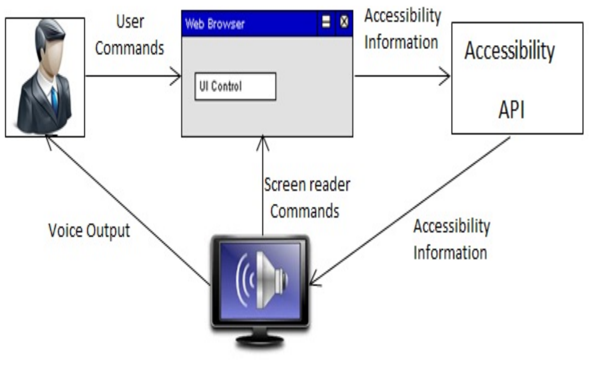

Figure1. Communication scheme of ARIA.

This particular accessible web application will use client side scripting language such as JavaScript to capture the keyboard shortcuts which are prompted by the screen reader and have some business logic written behind them. Ultimately, an acknowledgement is given to the user for every action or every change taking place on the screen; thus creating a virtual image about the overall layout of the web application in the user's mind.

4 CONCLUSION

We can see from this paper that there is immense potential in harnessing the united power of WAI-ARIA and on-screen readers to assist visually impaired users. This makes it possible for visually impaired users to use web applications with ease ultimately helping them to compete at equal levels with a normal user.

Thus, the above proposed solution can prove to be a major contribution in the field of Human Computer Interaction.

REFERRENCES

[1] W3C, WAI-ARIA Overview. W3C-WAI 09 2010: http://www.w3.org/WAI/intro/aria.

[2] [ARIA-PRACTICES] WAI-ARIA Authoring Practices. J. Scheuhammer, M. Cooper, L. Pappas, R. Schwerdtfeger, Editors, W3C Working Draft.

[3] Prof Dr. Helmut Krueger, Clausiusstr, Swiss Federal Institute of Technology. Zurich "Introduction to Human-Computer Interaction" 1998

[4] WAI-ARIA 1.0 Primer. W3C Working Draft 16 September 2010: http://www.w3.org/TR/wai-aria-primer/

- *Yogesh Lele, Maharashtra Institute of Technology, Pune, India, E-mail: lele.yogesh@gmail.com*
- *Chirag Kariya, Maharashtra Institute of Technology, Pune, India, E-mail: kariyachirag27@gmail.com*
- *Nupur Kale, Maharashtra Institute of Technology, Pune, India, E-mail: nupurkale@gmail.com*
- *Aarti Nandgaonkar, Maharashtra Institute of Technology, Pune, India, E-mail: aarti.10n@gmail.com*
- *Prof. Madhura Phatak (Guide) , Maharashtra Institute of Technology, Pune, India, E-mail: madhura.phatak@mitpune.edu.in*