Architecture of a Web Browser for Visually Handicapped People

Ritwika Ghose, Tirthankar Dasgupta, Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
West Bengal, India
ritwika.ghose@gmail.com, iamtirthankar@gmail.com, anupambas@gmail.com

Abstract—Internet has brought about an incredible improvement in human access to knowledge and information. However, blind people face difficulties in accessing these text materials. Web browsers for the visually handicapped people in the past have been limited to converting documents to Braille or speech, or extracting text and filtering. However, the human aspects of web surfing for blind people have not been adequately addressed. This paper presents an architecture of an open source, light weight web browser that makes it easy for the visually handicapped people to surf the web. The proposed architecture allows a blind person to navigate any web content through simple speech commands and voice feedback to any keyboard operation. The browser will have an integrated text extraction engine that inspects the content of the page to construct a structured representation. The internal nodes of the structure represent various levels of abstraction of the content. This helps in easy and flexible navigation of the page so as to rapidly home into objects of interest. Finally, the browser is integrated to an automatic Text-To-Speech and Text-To-Braille transliteration engine that outputs the selected text in the form of speech and/or Braille.

Keywords-Web Browser for blind; content reorganization; text-to-speech; text-to-Braille

I. INTRODUCTION

With the advancement in technology and the enhancement of the World Wide Web, information of all genres has reached the doorstep of people all over the world. People can learn anything and access news from all over the world sitting right at their homes through the Internet. However, the visually handicapped people are deprived of this benefit.

The National Census of India has estimated around 21.9 million disabled people in the country [3]. Out of which more than 15 million people in India are blind. This is considered to be the highest among all other disabilities. Three out of every five disabled children in the age group of 0-9 years have been reported to be visually impaired in India [12]. Due to their inability in accessing information from written text documents, blind people face tremendous difficulties in accessing information through web. Thus, in order to provide proper information access and to bridge the communication gap between the visually impaired and the sighted community, the need to build some advance technologically supported systems are utterly essential.

Several attempts have been taken to build web browsers for the blinds [2, 5]. However, these browsers suffer from several limitations like,

- Most of the web browsers do not support any automatic speech recognizer that will enable blind users to navigate a web page through speech commands.
- Most of the web browsers do not support any speech output to any navigation related operations performed by a blind user.
- Most of the web browsers for blind are integrated with text-to-speech engines, however, very few browsers support text to Braille representation that may be used by blind users to archive the web document in printed form.
- Although many web browsers are integrated with Text-to-Speech technology. However, most of these browsers read out the information on a web page to the user in a sequential pattern. This creates problems during navigation to a blind person.

There are two aspects of web surfing – a) surfing the content of a page, b) navigation through these pages. When a sighted user views a page, he/she can get a brief idea about that page by a mere glance. The user can then look for his/her area of interest and then read that particular content or may decide to leave that page. He/she may choose to skip any amount of information in between to reach the desired area of interest. However, visually handicapped users may face a great disadvantage here. They will not have any idea about the current active page. A screen reader merely reads out the whole content of the page as is. But that is very inconvenient for the users as in that case they may not have any control over the information in the page and will have to listen to the screen reader all through as it reads out the whole page, to get an idea about what is present. The user should be given the flexibility to explore the page in his own desired fashion. However, without prior knowledge about the page, it is difficult for a blind user to do so.

If the page contains forms or any place which requires user input, a sighted user can just click on that area and enter the information. But a visually handicapped person does not have this advantage. So it is also required to inform him about what is desired and where to input the information.
As it is difficult for a visually handicapped person to identify the different links present in the page, navigation through the page or through different pages is very difficult for him. A sighted user can just see the links and get an idea about where it leads to from information around the link. He/she can then just click the desired link and reach his/her area of interest. In case the context is not what he expects, he can simply hit the back button after a quick glance. But that is not so easy for a visually handicapped person. All these problems have lead to the necessity of designing an improved browser for the visually handicapped people. In this paper, we present an architecture for the same.

This paper starts with our motivation and objective of designing the browser in section II. Section III introduces the state of the art in this field. In section IV we present a survey on the type of web pages. Section V describes the proposed architecture in detail. Finally section VI concludes the paper and gives an idea about the work in progress.

II. MOTIVATION

Web pages are almost always specifically designed for sighted people. The main aim of web page designers is to convey information to Web surfers in a manner that is both attractive and convenient. However such organization of a page is not necessarily appropriate for a blind person. The main motivation of the architecture is to address this issue.

The objective of designing a special browser is to develop a framework, with the required toolset, to enable sightless people to browse the web. It is necessary to integrate multimodal input mechanisms with the system to simplify input mechanisms. Output should be converted to speech or Braille.

III. PREVIOUS WORK

A number of attempts were made worldwide to build web browsers and enhance usability of web for the visually handicapped people. In this paper we have thoroughly studied some of the most widely used web browsers for the blinds. Further, we have compared each of the browsers based on their key GUI features. Based on our study we consider the following GUI features to be essential for a web browser for the blinds.

- Open source architecture
- Text to speech output
- Navigations through voice feedbacks
- Speech recognition
- Text extraction and representation
- Text to Braille conversion
- Voice feedback for keyboard operations
- Mouse based gesture recognition

Next, we will discuss some of the commonly used web browsers for the visual impaired people.

Among the early attempts, voice input and input for surfing [1] was adopted for the visually handicapped people. IBM’s Home Page Reader [2], among others, presents the web page in an easy-to-use interface, and converts the text to speech, having different gender voices for reading texts and links. But the disadvantage of this is that the developer has to design a complex new interface for the complex graphical web pages to be browsed and for the screen reader to recognize.

<table>
<thead>
<tr>
<th>Features</th>
<th>WebbIE</th>
<th>eGuide Dog</th>
<th>Shruti Drishti</th>
<th>IBM Home Page Reader</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text to speech</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Speech recognition</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Voice feedback</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Text extraction and filtering</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Hierarchical representation</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Text to Braille</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Text to speech</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Since just voice input and output did not solve the problem of usability of the web for the visually impaired people, attempts to re-structure web pages gained popularity. [3] puts forward a simple browsing solution, which divides a web page into two dimensions. This greatly simplifies a web page’s structure and makes it easier to browse. But, due to the complexity and diversity of web pages, this system did not work as all web pages could not be simplified. Another web browser [4] generated a tree structure from the HTML document through analyzing links. Though this attempted to structure the pages that are linked together to enhance navigability, this did not prove very efficient for surfing. Furthermore, it did not handle issues regarding navigability and usability of the current page itself. The WebbIE browser [5] extracted the text, removed and used alt names for images, and represented the page as plain text for making it easy for any screen reader to present to the user. However, even this browser did not enhance navigability. Another browser developed for the visually handicapped people was eGuideDog [6] which had an integrated TTS engine. This system applies some advanced text extraction algorithm to represent the page in a user-friendly manner. However, still it did not meet the required standards of commercial use. [7] Describes an
accessibility Kit for blind people using a new language Blind Markup Language (BML).

On a different development, ShrutiDrishti [8], a web browsing system covered some new aspects like conversion of text to Braille. User can browse the web sites with single key input /minimum key combinations and provides a very user-friendly interface. BrailleSurf [9] also translated the content of the screen to Braille. This browser implemented simplification, i.e. removal of images, etc., rephrasing, i.e. adding labels before links, restructuring the page and use of different reading strategies, like extensive reading, or reading only links. Table 1 shows the comparison of the different web browsers discussed in this section in a tabulated form.

IV. SURVEY ON TYPES OF WEB PAGES

Depending on the purpose and content of a web page, these can be classified in various ways. It is important to understand and capture these because the presentation of a page to a blind person would depend on its type. We give below a broad classification with this in mind. This classification is based on the various formats followed by web page designers to design the pages based on what they want to present.

- Single article page
- Multiple article page
- E-mail
- Search Engine
- Portals
- Blogs
- Forms
- Social Networking
- Forums
- Online shops and auction web site

It may be noted that all these pages may contain multiple frames, tables, JavaScript, servlets, etc. These also have to be identified and handled when the content of the web page is presented.

V. ARCHITECTURE

Our proposed system essentially consists of three different modules:

1. User Input Module,
2. Text Extraction and re-organization, and
3. Output Representation.

A. User Input Module

The user input module deals with the different input methods. Operations on the browser, which are usually performed by mouse clicks, can be performed by speech commands. A speech recognizer is integrated with the browser, which is trained to recognize all the commands. The system is speaker independent. Apart from browser operations, different navigational operations can also be performed by voice commands. For ease in data input using keyboard, an optional voice feedback for every key operation is provided by the system. Figure 1 shows the input module.

B. Text Extraction and Re-organization

The extraction of text and presenting it to a visually handicapped person has many difficult aspects to it. With innumerable web pages present in the web, there is a varied diversity in the type of the pages. A prototype hierarchical structure of a web page is illustrated in figure 2. From the figure we can observe that, a web page may contain more than one kind of contents like, links, images, advertisements, and animations. These contents may not provide valuable information to a visually impaired person. Further, the document structure of an email page is also different from other pages.

In order to handle such kind of diversity in web pages, it is required to do a set of pre-processing on the corresponding pages before presenting it to the user. For that, the whole structure of the page along with its contents has to be extracted. This can be performed using an HTML parser. Using the parser, the syntactic information of a page is extracted from the html tags. For example, the header tags give information about the content following it and divider tags divide the pages into section which can be used to isolate different portions of the page as convenient. After getting an idea about the format of the page, the whole information in the page can be divided into sections and then presented to the user in a format convenient to them.

From the content of the web page, it is required to get a summary of the information that is conveyed by that page. In order to obtain that, some keywords important to the text can be retrieved to guess about the content. This information can then be used for further aid in structuring the page and also to give feedback to the user. As there may be pages which contain different information, there can be many levels of the summary.
As mentioned before, web pages contain a number of links. These may vary a lot in context. For example, some links are contained in the text itself, which may lead to further details of any context in a new page, or it may be linked to a different portion of the same page itself. Some links are given as menus which may lead to a completely different context, or to a sub topic. In this manner, there can be a variety of purposes of the links in a page. A sighted user can just view the page, see all the links and get an idea of which link will lead to where. In addition, that person can click on the link, and decide on its importance by just a quick glance and may immediately decide to return to the previous page. This benefit is not available to a visually handicapped person. So in our case, a study of the links is required. Information about the page where a link leads to, can be obtained in two ways. Sometimes, a line or two preceding the link can give information about where it leads to. Otherwise, the link itself can be retrieved in background, and again, some keywords relevant to the content may be extracted and an outline of the link can be presented to the user. Also voice commands for entering or skipping links will be provided so that the user has a choice to navigate through the pages as he/she wishes.

<table>
<thead>
<tr>
<th>Document</th>
<th>Text</th>
<th>Link</th>
<th>Advertisements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>Text</td>
<td>Link</td>
<td>Image</td>
</tr>
<tr>
<td>Text</td>
<td>Link</td>
<td>Image</td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>Text</td>
<td>Link</td>
<td>Image</td>
</tr>
</tbody>
</table>

Figure 2: Hierarchical Representation of a Web Page

After obtaining the structure of the page, it is required to rearrange the page in a new format. Control is provided in such a manner that any of the new sections that are created can be reached by voice commands. The user will be provided information about a section from the summary, and then, he/she may choose to continue listening to that section, or may break out and move to another section. This new format will also be given a hierarchical structure. The main heading and sub-headings of the page will be given a certain preference, the various menus and links will be also be given different importance, but unnecessary information, like advertisements, may be dropped off from the structure. In this manner, a web page will be presented part-by-part to the user, so that he/she can choose to hear about any part as desired, or move to any other portion if the current portion is no longer of interest.

Images or animations or any graphics in a web page are unnecessary for a visually impaired person. So in that case, the alternate names of the images can be conveyed to the user. If alternate names are not present for any graphical structure, it is simply dropped. In case of forms or text boxes or any place which requires user input, special notification to the user has to be given. Proper commands will control the cursor position during user input.

C. Output Representation Module

The formatted text from the text extraction engine is finally sent as an input to the output representation module. Figure 2 shows the block diagram of the output representation module. The module has the following two key features:

- Text to Braille transliteration engine
- Text to Speech engine

Based on the user’s preference, the output representation module can represent the formatted text either in the form of Braille or in the form of speech. The transliterated Braille output can further be printed using a Braille printer.

1) The Text-To-Speech Engine:
A text-to-speech (TTS) engine is at the core of the speech output generation module. It converts strings of text to their corresponding human voice equivalent. In order to develop the proposed transliteration architecture, we used a text-to-speech engine called Shruti [10] that uses diphone concatenation to synthesize speech in Indian languages using. The Shruti text-
to-speech engine currently supports two Indian languages namely Hindi and Bengali.

We built a wrapper around the TTS engine so that it can be used by a number of applications including our transliteration system. The wrapper also provides a number of other vital features like

- Conversion of text in Unicode or iTrans to ISCII\(^1\) which is the native representation format of the TTS being used,
- Pronunciations of individual and conjugate characters.

2) **Text to Braille Transliteration:**

The input to the transliteration unit is either an English text or an Indian language text document. The text documents are either based on Unicode, ASCII or ISCII encodings. Apart from the English language, our present system can transliterate the two Indian language texts namely, Hindi, Bengali. As discussed in [11], different Indian language characters shares the same Braille representation. Thus the rules to transliterate one Indian language text can be simultaneously being applied to any other Indian language texts. The transliterated text generated from the Braille transliteration unit, can be easily embossed using a standard Braille embosser. We have applied the same transliteration algorithm that has been discussed in the literature [11].

VI. **CONCLUSION**

In this paper, the architecture of a special browser for the visually handicapped people has been described. The browser is currently under construction. The output module, i.e., Text-to-Speech and Text-to-Braille and the voice feedback for keyboard operation is complete. An Automatic Speech Recognition system is currently being trained with the voice commands. Being open source, future developments on the browser by different groups would be easy. The browser is expected to be more user-friendly and effective for the visually handicapped people and hopefully reducing the gap of information availability between the sighted and the visually impaired people.

**ACKNOWLEDGMENT**

The authors would like to thank Department of Science and Technology, Govt. of India for financially supporting this project and the members of Communication Empowerment Laboratory for an excellent environment for this work.

**REFERENCES**


\(^1\) Indian Standard Code for Information Exchange