Application of Content Adaptation in Web Accessibility for the Blind

Pauli P. Y. Lai
The University of Hong Kong
Pokfulam, Hong Kong
pauli@hku.hk

ABSTRACT
While many people enjoy surfing the Web without any difficulty, the blind people could only read webpage with the help of screen reader which reads aloud the whole webpage. This approach is quite time consuming for them to get to their interested content even though some screen readers allow tabbing through links or headings. It is proposed to adapt the webpage into different logical sections, each would be given a descriptive heading and a number so that the user can enter the number in order to get into the details of that section. The whole idea is like converting the webpage into an IVRS (Interactive Voice Response Systems) so that the blind people can access the webpage using mobile phone by hearing the index page and getting into details by pressing the corresponding number on the key pad.

Categories and Subject Descriptors
H.5.3 [Group and Organization Interfaces]: Web-based interaction

General Terms
Algorithms, Design, Human Factors

Keywords
Content Adaptation, Web Accessibility, Blind, Mobile Web Browser

1. INTRODUCTION
While many people enjoy surfing the Web for information and entertainment, some people in the world—who are unable to see—cannot take this advantage to some extent. To allow the Web being accessible by everyone including people with disabilities, this is the matter of Web accessibility. Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. It encompasses all disabilities that affect access to the Web, including visual, auditory, physical, speech, cognitive, and neurological disabilities [1]. But the focus of this paper is put on the people with visual disability. Approaches suggested to aid people who are completely blind are different from that for people with “low vision”. We are aiming at facilitating the access of blind users to the Web. For those who are completely blind, they can perceive webpage by alternative sensory channels such as auditory channel or through tactile devices. The most common way for them to access the Internet is to use a Web browser and text-to-speech software such as Microsoft Narrator and Mac VoiceOver [2]. But the problem of using these screen readers is that they usually start reading from the top and traverse the whole webpage sequentially. Imagine if the information interested to the user is located hundreds of lines after, the user has to wait for a long time until the screen reader finishes reading those hundreds of lines before it reaches the destination. Another problem is that some screen reader may read aloud layouts which are meaningless to the blind user. Images and videos are also inaccessible items except they provide descriptive text with alt attribute.

The Job Access With Speech (JAWS) system is one of the popular screen readers that presents the information as synthesized speech. It provides a number of features for quick access of information using Navigation Quick Commands. It also provides some information about the structure and organization of webpages by announcing Tables, lists, headings, and alternate text. However, it is still a kind of sequential presentation of information but with some shortcut keys provided for jumping around [3].

Doubtlessly and unquestionably, the blind can read webpages only by hearing and touching. Characters on the webpages can become sounds or Braille codes for the blind to read. But the way of display can be better than just translating the webpages from the top to the bottom sequentially. In this paper, it is proposed to re-organize the webpage according to the structure of its semantic content, group the related content into logical sections, and assign a number and a descriptive heading to each section. The webpage is then presented as a menu list of headings and the user just needs to enter the corresponding number of their interested section. The blind user would no longer be forced to hear hundreds of lines before reaching their destination.

For the input device used, we propose to use a simple keypad with at least 10 numbers and a few of function keys that allow users to input special instruction. With this simple keypad and the adapted list of menu headings, the user can listen to the menu lists and then enter the corresponding number in order to get into the details of the interested section. The whole idea is like converting the webpage into an IVRS (Interactive Voice Response Systems)
so that the blind people can access the webpage by hearing the index page and getting into details by pressing the corresponding number on the key pad.

In this paper, we will describe how to adapt an ordinary webpage for being “browsed” by the blind. We first reverse engineer a webpage by parsing the webpage to extract the semantic elements and their relationship. The relationships would then be used for logical section discovery. A logical section is a group of related semantic elements that exist as an independent unit. A number and a descriptive heading would then be assigned to each logical section so that the user is able to explore each section by just pressing the corresponding number.

The rest of the paper is organized as follows. Section 2 will present our reverse engineering process. Section 3 will discuss how to improve Web accessibility for the blind and section 4 will conclude our work.

2. REVERSE ENGINEERING PROCESS

In [4], reverse engineering is defined as the process of analyzing a subject system to identify the system’s components and their interrelationships and create representations of the system in another form or at a higher level of abstraction. To adapt this definition to our system, it is revised as the process of analyzing a Web document to identify the page elements and their interrelationships and create a model to represent the Web document.

According to Lum and Lau[5], a Web document is indeed composed of semantic elements, container elements and layout elements. Semantic elements are the elements that convey semantic information. These elements constitute the content of the webpage and provide information to the user. For example, the Text element, the Anchor element and the Image element all contribute some semantic information to the Web document.

Next, a model is built to represent the relationship between the semantic nodes. We would like to build a DOM-like model that is used to model the relationship between semantic nodes. We call it semantic-DOM. Here we use an example to illustrate our reverse engineering process. First, the HTML code of a webpage from www.yahoo.com is shown in Fig. 1 is passed to the system for parsing. After the process of relationship analysis, we grouped all the related elements into a logical section. The semantic DOM tree as shown in Fig. 2 is generated to represent the groups of logical sections.

![Figure 1. A webpage from www.yahoo.com is used to illustrate our reverse engineering process.](image-url)
Moreover, the user would be told of the function, if there is any, of a particular section. For example, the heading would be preceded with “Form” if the section is a form. If there is a hyperlink associated with the heading, the heading would be preceded with “Link”. If the section contains the elements with descriptive relationship, the heading would be preceded with “Description”. In that way, the user would be informed what kind of section they are entering.

Here is the heuristic for generating headings of menu list. After obtaining the semantic-DOM tree from the step in section 2, section headings and menu headings would be generated respectively. We adopt a bottom-up approach to generate section headings first. For each section from bottom to top, if the section contains only one child in the immediate next level, the content of the child would become the heading of the section, preceding with the function of the child, if any, such as “Link”, “Enriching”, “Description”, etc. Otherwise, if the section contains many children in the immediate next level, the content of the first child would be adopted as the heading of the section, preceding with the function of the group of children, if any. Since only the content of the first child is used, the heading should be followed by some indication such as “…” or “etc” so as to let the user know there are more children in the group. Fig. 3 illustrates the section heading generated for the semantic DOM tree shown in Fig. 2. Next, after all the section headings have been assigned, the headings for the menu list would then be generated. This time we adopt a top-down approach to generate the menu headings. Starting from the root, we assign the menu headings level by level. For each section or semantic node in each level, a number is assigned sequentially. The menu heading for semantic node is the content of the node preceding with its function such as “TEXT”, “LINK”, or “Image”. On the other hand, the menu heading for the section node is generated by concatenating the section headings of the children in the immediate next level. Fig. 4 shows the menu headings generated from Fig. 3.

In section 2, the logical sections have already been identified. The next step is to assign a descriptive heading for each section so that these headings are used to construct the index page. For the section that contains the elements with enriching relationship and descriptive relationship, the enriching heading and the description node can be adopted directly as the headings for the enriching details and descriptive details group respectively. But for the section of parallel elements, there is no such enriching heading that can be adopted directly. In the current stage, we only use the first element of the group for the heading. This could be improved in future by applying some algorithm to extract an appropriate title for the parallel group.

**3. IMPROVED WEB ACCESSIBILITY FOR THE BLIND**

The above reverse engineering process can be applied to improve Web accessibility for the blind. From my initial study, I found that the blind people perceive webpage by listening to the screen reader which reads aloud the whole webpage. This approach is quite time consuming for them to reach their interested content even though they can use keystrokes to tab through, for example, the links or headings. It is proposed to adapt the webpage into different logical sections using the above reverse engineering process, each section would be given a descriptive heading and assigned a number so that the user can enter the number in order to get into the details of that section. There may be several levels of abstraction for the sections. The whole idea is like converting the webpage into an IVRS (Interactive Voice Response Systems) so that the blind people can access the webpage by hearing the index page and getting into details by pressing the corresponding number on the keypad. It could be used on desktop computers, laptop computers and all kinds of mobile devices provided that there is an audio output for reading aloud the menu list of the adapted webpage and a keypad for inputting the corresponding number of interested section.

![Figure 2. An abstract of a semantic DOM tree that is generated to represent the groups of logical sections.](image)

![Figure 3. Section headings generated for the semantic DOM tree in Figure 2.](image)
There may be several levels of abstraction for the logical sections. An example is illustrated in Fig. 5. It requires intelligence to decide the number of levels and the number of menu items for each level. Reasonably, the number of menu items in each level should not exceed 10 since there are at most 10 numbers in a keypad. Some people may argue that the user may enter 2-digit number for the option but it is neither good for pressing two digits for one option nor listening to a menu list with more than 10 options. If it exceeds 10 inevitably, some sections may have to be grouped together and the corresponding section heading is constituted by concatenating the headings of those grouped sections. It also requires intelligence to create a heading with reasonable length. Theoretically, the number of levels of abstraction could be as many as possible until the ultimate content is reached but too many levels would lead to lengthy time in reaching destination. It is a tradeoff to balance in between the number of levels and the number of sections in each level. Fig. 5 shows an example sequential flow of selecting different number at different level. The ultimate content is reached at level 6 in which option 3 has been selected in level 5, it would then go to the corresponding hyperlink. Note that when it goes to a hyperlink of a webpage, a new page would be adapted and a new menu list of this webpage would be read aloud.

To improve user friendliness, some additional headings may be inserted to the menu list or some functional keys in the keypad would be adopted for special use. For example, the user may press “*” key to go back to the previous page and “#” key to read again all the options.

4. CONCLUSION
In conclusion, we have proposed a reverse engineering process to reverse engineer a webpage so that the semantic elements are extracted and the relationships between them are identified. With these relationships, the logical sections that group related semantic elements together can be discovered. A model named semantic DOM has been proposed. The semantic DOM is a DOM-like model that is used to represent the groups of logical sections. The discovery of logical sections will be useful in improving Web accessibility for the blind. It is anticipated that if this mobile Web browser for the blind is successfully implemented and launched, the blind people would be able to surf the Web using their fingers and ears. They would also be able to access the Web anytime and anywhere using their mobile phones. This would bring much convenience to the blind and would change their life to be more fruitful thereafter.

5. ACKNOWLEDGMENTS
I would like to thank my supervisor, Prof. Francis C.M. Lau, for his guidance and support. Also, I would like to thank my senior fellow, Mr. Vincent W.Y. Lum, for his comments and legacy work.

6. REFERENCES