

# Generating Efficient Labels to Facilitate Web Accessibility

Leo Spalteholz  
Department of Electrical and  
Computer Engineering  
University of Victoria  
Victoria, BC, Canada  
V8W 3P6  
leos@ece.uvic.ca

Kin Fun Li  
Department of Electrical and  
Computer Engineering  
University of Victoria  
Victoria, BC, Canada  
V8W 3P6  
kinli@ece.uvic.ca

Nigel Livingston  
University of Victoria Assistive  
Technology Team  
University of Victoria  
Victoria, BC, Canada  
V8W 3P6  
njl@uvic.ca

## ABSTRACT

For many users with a disability it can be difficult or impossible to use a computer mouse to navigate the web. An alternative way to select elements on a web page is the label typing approach, in which users select elements by typing part of the label. In most cases, these labels are specified by the page authors, but some selectable elements do not have an obvious textual description, thus requiring that a label be generated. The set of element labels on a web page must be both efficient to select by text input and meaningful to the user. This paper discusses our approach to this problem, using page structural analysis and user history to determine important elements of a page, and then matching this information with the efficiency model of the input device.

## Categories and Subject Descriptors

H.5.2 [Information interfaces and presentation]: [User Interfaces – Input devices and strategies.]; K.4.2 [Computers and society]: [Social issues – Assistive technologies for persons with disabilities.]

## General Terms

Human Factors

## Keywords

Web Accessibility, Labeling, Web Navigation, Single Switch

## 1. INTRODUCTION

For people with physical disabilities, especially those whom are only able to operate very low bandwidth input devices, the task of navigating on the world wide web can be very tedious. This problem is particularly acute for those users who depend on a single switch input device and thereby do not have many options for browsing and conducting business using the world wide web. The primary task in navigating the web is to select elements on a web page, such as hyperlinks, text fields, or various form controls. Existing solutions to this problem have taken the approach of linearly iterating through every selectable element on the web page, briefly pausing on each one, and selecting it when the user activates a switch. This approach, while simple

to implement, suffers from several problems including poor scaling to large numbers of elements, unpredictable scanning sequences, and high error cost. To address these issues, we have previously proposed an alternative approach based on selecting elements by text entry instead of direct selection [2]. In this approach, each selectable element is assigned a label, and the user selects an element by typing a small substring of this label. In most cases, this allows any given element on a page to be selected with one or two characters (produced by a virtual keyboard).

Since activating an input device often requires significant effort from a disabled user, a primary goal when generating element labels is to make their selection as easy and efficient as possible. A secondary goal is to make the text as descriptive of the element as possible to lessen cognitive load during the selection process. This paper discusses our approach to meeting these goals using the web page structure and user history to determine salient items on a web page, then matching this information with the efficiency model of the input device. We are currently in the process of implementing our system as an extension to the Firefox web browser. Our system overview is shown in Figure 1 and explained in the following sections.

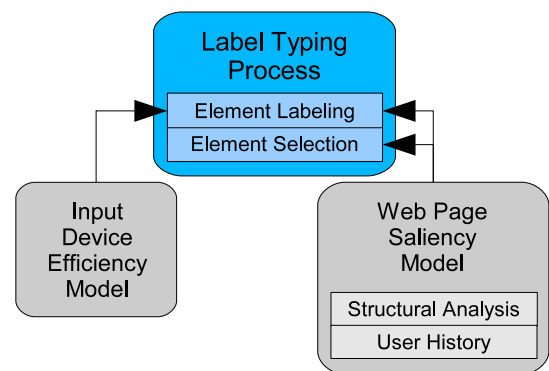


Figure 1: System Overview.

## 2. DETERMINING ELEMENT SALIENCE

Web pages usually contain some set of selectable elements that are more important than others on the same page. These elements are often hyperlinks in the navigation bar, pointing to main sections of the web site. We have two sources of information for determining which elements are important to the user and thus should be made efficiently se-

lectable: the user’s personal browsing history and the structure of the web page. To determine important hyperlinks, we can either perform a static analysis of the web page or retrieve linked pages to find strongly connected pages on the same site [1]. While analysis of the web page’s structure is likely to identify elements that the page’s author has deemed important, it does not take into account the browsing history of the user. For this, the system (at its most basic) keeps track of selectable elements that have been activated in the past by the user and prioritizes these for the labeling and selection process.

### 3. INPUT DEVICE EFFICIENCY

To optimize the labeling process for selection efficiency, we need information about the cost of generating any given letter, number, or symbol with the user’s input device. Since the primary focus of this web access method is persons using a single switch to communicate, we will examine the efficiency model of a typical virtual scanning keyboard used by many single switch users. Characters are arranged in a grid, and selected in two stages, first selecting the correct row and then the column. The average time cost for selecting an element in a two stage scanning keyboard can be calculated as:

$$t_{avg} = \left(\frac{\sqrt{n}}{2} + \frac{\sqrt{n}}{2}\right) \times T = \sqrt{n} \times T$$

Where  $n$  is the number of characters in the virtual keyboard and  $T$  is the scanning delay.

Since only alphanumerics are necessary for the label typing approach, we require that the keyboard contains at least 36 elements. Thus, the average cost to type one character is  $t_{avg} = \sqrt{36} \times T = 6 \times T$  and two switch activations. Furthermore, characters in the virtual keyboard can be ranked according to their time cost, with those in the top left corner being fastest to type (two clicks, no wait), and those in the bottom right being slowest (two clicks,  $12T$  wait). We use this information on relative character cost when generating the labels for elements that do not have an existing textual label. Elements with a higher priority (Section 2) will be assigned labels starting with the minimum cost characters that are not already in use on the page.

### 4. MEANINGFUL ELEMENT LABELS

All selectable elements on a web page can be classified as either having author-specified labels or requiring a label to be generated. Elements with user-visible textual descriptions are text hyperlinks, radio/check/push buttons, text fields with external label tags, and images with alternative text. The second category, elements without an author-specified textual description such as text input fields, are more difficult to label. When a selectable element does not have an obvious label, one approach is to generate a label from metadata such as the name of the image file or the name of the element type. Some simple heuristics can be used to determine if the image name or element ID attribute is likely to be human readable (English word substring, ratio of letters to other symbols, etc). To obtain more meaningful labels for image links at the cost of increased label generation time, it is possible to fetch the linked page and extract data from it, either from the page’s title or by more advanced keyphrase identification [3].

## 5. OPTIMIZING ELEMENT SELECTION

With information about which items are likely to be more important to the user, the selection process can be optimized. The example web page shown in Figure 2 contains several hyperlinks starting with the letter “c”. Without link prioritization, all of them would be highlighted in yellow after the user typed “c”, requiring additional letters to differentiate them. In this example, our process for identifying important links has determined that the “Contact” link is more likely to be selected by the user, and has therefore made it the default selection (highlighted in green). The user may select this link without typing any more letters by selecting the equivalent of the enter key in their virtual keyboard. At any time when there are multiple matches (even when nothing has been typed) the system selects the most likely candidate element as a default selection that can be immediately activated.

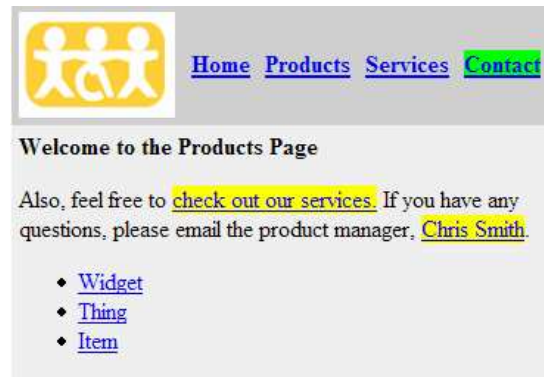


Figure 2: Example of Element Prioritization.

## 6. CONCLUSIONS

Building on our previous work on the label typing system of navigating web pages for users with physical disabilities, we have presented techniques to improve the efficiency of the selection process. By intelligently identifying important elements in a web page’s structure, and taking into account a user’s past browsing behavior, we can prioritize certain elements for faster selection. Extracting information from linked pages allows us to create more meaningful labels for elements that do not have one specified by the page’s author. The implementation of our system is progressing in the form of an extension to the Firefox web browser.

## 7. REFERENCES

- [1] Z. Liu, W. Ng, and E. Lim. An automated algorithm for extracting website skeleton. In *Proceedings of the 9th International Conference on Database Systems for Advanced Applications*, pages 799–811, 2004.
- [2] L. Spalteholz, K. F. Li, and N. Livingston. Efficient navigation on the world wide web for the physically disabled. In *Proceedings of the 3rd International Conference on Web Information Systems and Technologies*, pages 321–326, Mar. 3-6, 2007.
- [3] P. D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.