A Web Browsing System based on Adaptive Presentation of Web Contents for Cellular Phones

Yuki ARASE† Takuya MAEKAWA† Takahiro HARA† Toshiaki UEMUKAI‡ Shojiro NISHIO†

†Dept. of Multimedia Engineering, Grad. Sch. of Information Science and Tech., Osaka Univ.
‡KDDI R&D Laboratories Inc.
{arase.yuki,t_maekawa,hara,nishio}@ist.osaka-u.ac.jp, to-uemukai@kddilabs.jp

ABSTRACT

Cellular phones have already been widely used to access the Web. However, most existing Web pages are designed for desktop PCs, and thus, it is inconvenient to browse these large Web pages on a cellular phone with a small screen and poor interfaces. Users who browse a Web page on a cellular phone have to scroll the whole page to find an objective content, and then, have to scroll within the content in detail to get useful information. In this paper, we propose a novel browsing system to break off these burdensome operations by adaptively presenting Web contents according to their characteristics.

Categories and Subject Descriptors

H.4.3 [**Communications Applications**]: Information browsers; H.5.2 [**User Interfaces**]: User-centered design—*Screen design*

General Terms

Human Factors

Keywords

Cellular phone, Web browsing, Overview, Adaptive presentation

1. INTRODUCTION

Cellular phones have been widely used to accsess the Web due to their progress in processing and communication facilities. However, most existing Web pages are designed for desktop PCs, and thus, it is inconvenient to browse such large pages by using cellular phones. Since cellular phones only have a small display and poor interfaces, users have to perform numerous operations for scrolling a whole page.

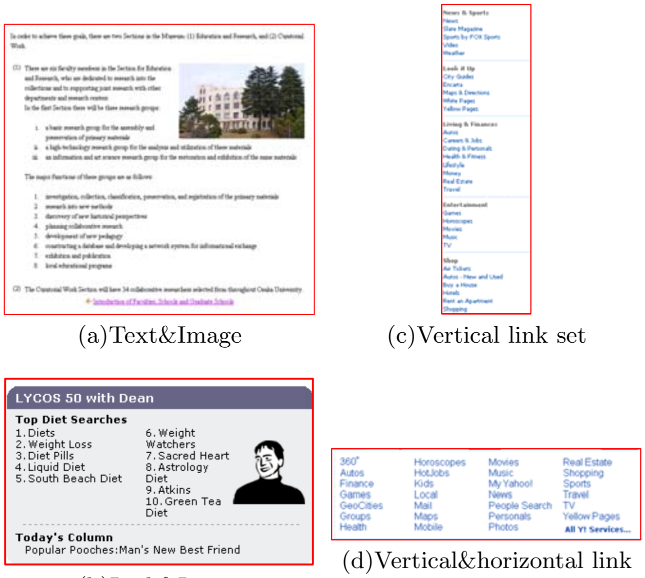
Generally, a Web page is composed of a large number of different components[1, 3, 5], each of which is an information block such as a site directory, news, and a search form in the top page of a portal site. Figure 1 shows an example, where each block enclosed with frame is a component. Due to such a structure in Web page, users usually browse a Web page in the following two steps. First, a user looks over a page to find an objective component, then reads information within the component in detail. Because of the small display of a cellular phone, the user has to scroll the whole Web page for a long time to find the objective component. Furthermore, the user has to scroll within the component to read the information in detail. To provide comfortable Web browsing using cellular phones, these burdensome scroll operations have to be reduced. In [2], Chen et al. proposed a system that provides users an overview, which is a scale-down image of a Web page. When a user selects a content which he/she wants to read in an overview, the area around the selected content is zoomed in. By doing so, the user can find an objective component from the page with fewer scroll operations. However, the system proposed in [2] did not focus on reducing complicated scroll operations to read within components. Besides, users' behaviors in reading Web components differ depending on the components' characteristics. Thus, it is effective to present each component adapted to the users' reading behaviors.

Figure 1: A Web page and components

In this paper, we describe design and implementation of a Web browsing system which adaptively presents components of Web pages according to their characteristics to reduce users' scroll operations. Our system displays the overview of a page so as to reduce time-consuming scroll operations to find an objective component from the page. In addition, this system presents the component adapted to its characteristics so as to reduce burdensome scroll operations to read the component.

The reminder of the paper is organized as follows: In section 2, we describe the components' classification and a preliminary experiment. Section 3 describes the design and implementation of our system. In section 4, we further discuss our proposed system. Finally, we give conclusion and remarks about the future work in section 5.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*W4A at WWW*2006 23-26 May 2006, Edinburgh, UK
Copyright 2006 ACM 1-59593-281-x/06/05 ...$5.00.



(a)Text&Image (c)Vertical link set

(b)Link&Image (d)Vertical&horizontal link set

Figure 2: Examples of component classes

2. COMPONENT CLASSIFICATION AND PRELIMINARY EXPERIMENT

As described in section 1, Web pages are composed of various components. In this section, we explain components' classification and a preliminary experiment to determine an appropriate presentation for each component class.

2.1 Components classification

We consider that components have common characteristics according to their contents. We have checked fifteen typical Web sites such as portal and news sites and found that components are classified into the following six categories according to their contents:

- *Text*: contains only texts.
- *Text&Image*: contains both texts and images associated with the texts. Figure 2(a) shows an example.
- *Image*: contains only images.
- *Link&Image*: contains both a set of links and images. Figure 2(b) shows an example.
- *Vertical link set*: contains a set of vertically listed links. Figure 2(c) shows an example.
- *Vertical&horizontal link set*: contains a set of both vertically and horizontally listed links. Figure 2(d) shows an example.

2.2 Reading style of each component class

Users' reading behaviors change according to components' class, which we described in subsection 2.1. For example, a user thoroughly reads each sentence in a "Text" component, while he/she looks at images as well as reading texts in a "Text&Image" component. Besides, a user searches and selects an objective link or a link of some interest in a "Vertical link set" or "Vertical&horizontal link set" component.

By adaptively presenting a component according to the users' reading behavior for each component class, burdensome scroll operations can be reduced.

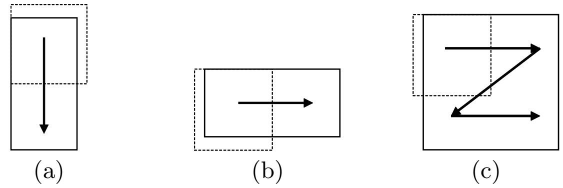


Figure 3: Scroll paths

Table 1: Scroll patterns

Speed/Zoom ratio	× 1	× 2	× 3
20pix/sec	pattern 1	pattern 2	pattern 3
50pix/sec	pattern 4	pattern 5	pattern 6
100pix/sec	pattern 7	pattern 8	pattern 9

2.3 Preliminary experiment

We conducted a preliminary experiment to determine how to present a component according to its belonging class and characteristics. We consider auto-scrolling as an effective approach because it can reduce users' burdensome scroll operations within a component. In this experiment, we examined the effectiveness of auto-scrolling and its appropriate setting.

2.3.1 *Experimental setting*

Sixteen subjects were asked to read six components using auto-scrolling, where each component belongs to each of the six component classes described in subsection 2.1. Our experimental system determines a scroll path for each component based on the component's shape. Specifically, when the component's height is higher than that of the cellular phone's display and the component's width is narrower than that of the display, the scroll path is set in vertical direction as shown in Figure 3(a). On the contrary, when the component's height is lower than that of the display and the component's width is broader than that of the display, the scroll path is set in horizontal direction as shown in Figure 3(b). When both the component's height and width are larger than those of the display, the scroll path is set in zigzag as shown in Figure 3(c). Here in Figure 3, a rectangle with solid line represents a component's shape, a rectangle with dotted line represents a cellular phone's display, and an arrowhead represents a scroll path with direction. In this experiment, we examined nine patterns for auto-scrolling with different speed and zoom ratio, where the subjects selected the best pattern for auto-scrolling for each component class.

2.3.2 *Experimental result*

We show the result of the preliminary experiment. Due to the limitation of space, we describe only a short summary. From the result, it is shown that auto-scrolling is not suitable for "Text" and "Text&Image" components, because the speed to read sentences differs among individuals. It is also shown that subjects who are reading a "Text&Image" component by auto-scrolling could not look at images within the component when they wanted. On the other hand, we confirmed that auto-scrolling is suitable for "Image", "Link&Image", "Vertical link set", and "Vertical&horizontal link set" components. However, some subjects said that it was hard to read these components if they were presented in zigzag. For "Image" components, most subjects evaluated that pattern 4 is the best, while for "Link&Image", "Vertical link set", and "Vertical&horizontal

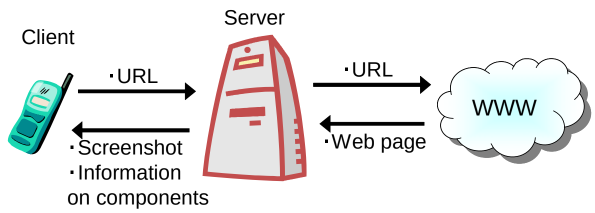


Figure 4: System design

link set" components (in the following, we call them "Link set" components), subjects' evaluations are split across, some thought pattern 1 the best and the others thought pattern 4 the best. This seems due to differences in the subjects' style for searching links. Subjects who liked pattern 1 thoroughly read each text associated with a link, and the others skimmed through the component.

3. SYSTEM DESIGN AND IMPLEMENTATION

Based on the result of the preliminary experiment, we designed and implemented a system that offers adaptive component presentation in a Server/Client architecture. The server was implemented on a PC with Windows XP, coded with Visual C# and PHP (Hypertext preprocessor). The client was implemented on a cellular phone (SH900i), which is equipped with i-appli developed by NTT Docomo, coded with Java. Within the 240 × 320 [pix] display of the cellular phone, i-appli can only use a 240 × 240 [pix] area.

Figure 4 shows the design of our system. In response to the client's request, the server receives the corresponding Web page from a WWW server, and makes a screenshot of the page. Then, the server extracts components from the page, determines the components' classes, and then, sends the screenshot, information on the components' classes, and information on the components (the location, height, and width) to the client. The client displays the overview of the requested page, which is created from the screenshot and the information on the components received from the server. When the user selects a component on the overview, the client performs an adaptive presentation of the component. In the following, we describe the details of our implemented system.

3.1 Overview

A user can select a component which he/she wants to read on the overview of a Web page. An overview is a scale-down page that fits the display size of the user's cellular phone (Figure 5(a)). The component that the user is watching with attention, i.e., that the pointer is currently focused on, is automatically zoomed in. The user can further zoom in the component by pressing a certain key of his/her cellular phone (Figure 5(b)). In addition, the user can perform the following operations;

Selecting of a component: A user can select a component to read by pressing a certain key of his/her cellular phone, then the selected component is presented adaptively.

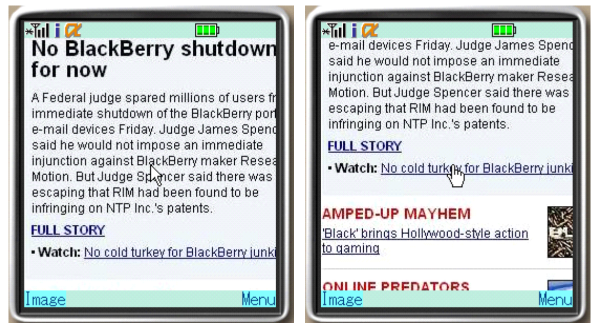
Back: A user can go back to the Web page that the user read before the current page.

Forward: A user can go forward to the page that the user read before going back to the current page.

3.2 Adaptive presentation

(a) Overview (b) Zooming in

Figure 5: Overview



(a) Manual scroll (b) Selecting a link

Figure 6: Adaptive presentation of "Text" components

In this subsection, we describe how our system adaptively present components.

3.2.1 *"Text" component*

In our system, users read "Text" components by scrolling manually, because it has been shown from the preliminary experiment that the speed to read sentences differs among individuals. Figure 6(a) shows a screenshot of a cellular phone displaying a "Text" component, where the arrowhead in the center is a pointer. The shape of the pointer changes when the pointer is on a letter string of a link as shown in Figure 6(b), where the user can select the link. In addition, the system offers users "Jump" function. The user can jump to the top of the next sentence by pressing a key of the cellular phone when he/she finishes reading the current sentence.

3.2.2 *"Image" component*

An "Image" component is presented by auto-scrolling in speed of 50 [pix/sec] which is determined by the preliminary experiment. The preliminary experiment also showed that auto-scrolling in zigzag is hard for users to follow. Therefore, in our system, if the size of a component is larger than that of the cellular phone's display, the component is zoomed out until it's height or width fits the display's height or width. In addition, users can stop auto-scrolling, then he/she scrolls manually, and can select a link in the same way as "Text" components.

3.2.3 *"Text&Image" component*

The preliminary experiment showed that users cannot read "Text&Image" components comfortably by auto-scrolling because they cannot watch images within the components at the time they wish. Therefore, in our system, users read sentences within "Text&Image" components by scrolling manually, and they can switch over images within the components by pressing a key when he/she wants to watch them. That is, the users can switch over the images and sentences in rotation by pressing the key consecutively. By doing so, the users can watch images associated with sentences when they wish, and then can go back to the sentences. In addition, the system offers "Jump" function, and users can select a link in the same way as "Text" components.

3.2.4 "Link set" component

"Link set (Link&Image, Vertical link set, and Vertical& horizontal link set)" components are presented by auto-scrolling because the effectiveness was confirmed by the preliminary experiment. In doing so, the system determines the speed and zoom ratio of auto-scrolling according to contents of the component. First, the speed is determined by using the following formula:

$$Speed[\text{pix/sec}] = \frac{\alpha}{ID \cdot Breadth}. \quad (1)$$

Here, α [characters/sec] is the number of characters which human can recognize per one second in auto-scrolling, ID (information density) [characters/pix^2] is the number of characters per unit area, i.e., 1 [pix^2], supposing that all the characters within the component are uniformly distributed. $Breadth$ [pix] is the width (height) of the component in the orthogonal direction with that of auto-scrolling, represented by the following formula:

$$Breadth = \begin{cases} component's\ width & (vertical\ scrolling) \\ component's\ height & (horizontal\ scrolling). \end{cases}$$

Consequently, α is given by the following formula:

$$\alpha = Speed \cdot ID \cdot Breadth.$$

We determined the value of α respectively for each of "Link set" components, using ID, $Breadth$, and auto-scrolling speed (20 [pix/sec] or 50 [pix/sec], which the users liked in the preliminary experiment) which were obtained by the preliminary experiment. Then, by using these parameters' values, the auto-scrolling speed is calculated by using formula (1). The preliminary experiment also showed that there are two different styles for searching an objective link; thoroughly reading link texts and skimming through the component, and these styles require different auto-scrolling speeds. Accordingly, a user should select an appropriate value of α adapting to his/her reading style so that he/she can set an appropriate scrolling speed.

Next, the path and zoom ratio of auto-scrolling are determined as follows. In our system, if the width (height) of the component is larger than the height (width), the path is set in horizontal (vertical) direction. If the size of component is larger than the cellular phone's display, the component is zoomed out so as to avoid auto-scrolling in zigzag.

In addition, the system offers functions of rewind and fast-forward. Moreover, users can stop auto-scrolling and scroll manually anytime they want, and then can select a link in the same way as "Text" components.

4. DISCUSSION

4.1 Path selection for auto-scrolling

In our current implementation, the path of auto-scrolling is determined by comparing the width with the height of the component. This is effective when auto-scrolling is performed in one direction, vertical or horizontal direction, while there is a serious problem for auto-scrolling in zigzag. Specifically, the information within the component is sometimes presented by snatches in auto-scrolling in zigzag. For example, let us suppose a component that includes a large number of links with letter strings enumerated in vertical direction and the length of the strings are longer than the width of the cellular phone's display. This component is presented by auto-scrolling in vertical direction, however, the letter strings of the links are presented by snatches. Thus, our system should be extended to determine a path of auto-scrolling considering the direction and lengths of letter strings of links, as well as the existence of images and the width and height of a component.

4.2 Decision of component's class

Our system sometimes misjudged the component's class, for a "Vertical&horizontal link set" component as "Link& Image", because it deals equally with large images and small images, e.g., icon images. To solve this problem, it should be effective to classify images within a Web page according to their roles [4] and deal with them in different ways.

5. CONCLUSION

In this paper, we proposed an adaptive presentation system of Web contents for Web browsing using cellular phones. This system reduces successive scroll operations to find an objective component by providing the overview of the page. In addition, the system presents the component adaptively according to it's component class, and thus, users can read it comfortably. We confirmed the effectiveness of our system by using a simple user experiment. Due to the limitation of space, we omit the details in this paper.

As part of our future work, we plan to address an issue to determine a better path of auto-scrolling. Moreover, we plan to conduct detailed user experiments and verify an appropriate speed for auto-scrolling.

Acknowledgments

The authors wish to thank Dr. Shigeyuki Akiba, President & CEO of KDDI R&D Laboratories Inc. for his continuous support for this study. This research was partially supported by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

6. REFERENCES

[1] J.-R. W. D. Cai, S. Yu, and W.-Y. Ma: "Extracting content structure for web pages based on visual representation," Proc. Asia Pacific Web Conference (APWEB2003), pp. 406–417, Apr. 2003.

[2] Y. Chen, W. Ma, and H. Zhang, "Detecting Web page structure for adaptive viewing on small form factor devices," Proc. WWW'03, pp. 225–233, May 2003.

[3] D. W. Embley, Y. Jiang, and Y. K. Ng: "Record-boundary discovery in web documents," Proc. ACM SIGMOD'99, pp. 467–478, June 1999.

[4] T. Maekawa, T. Hara, and S. Nishio: "Image classification for mobile Web browsing," Proc. WWW'06, to appear, May 2006.

[5] Y. Yang and H. Zhang: "HTML page analysis based on visual cues," Proc. ICDAR2001, pp. 859–864, Sep. 2001.