Proc. of IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom 2006), pp. 22-35 (Mar. 2006)

A Collaborative Web Browsing System for Multiple Mobile Users

Takuya Maekawa, Takahiro Hara, Shojiro Nishio
Dept. of Multimedia Eng. Sch. of Information Science and Tech., Osaka Univ.
{*t_maekawa,hara,nishio*}*@ist.osaka-u.ac.jp*

Abstract

In mobile computing environments, handheld devices with low functionality restrict the services provided for mobile users. We propose a new concept of collaborative browsing, where mobile users collaboratively browse web pages designed for desktop PC. In collaborative browsing, a web page is divided into multiple components, and each is distributed to a different device. In mobile computing environments, the number of handheld devices, their capabilities, and other conditions can vary widely amongst mobile users who want to browse content. Therefore, we developed a page partitioning method for collaborative browsing, which divides a web page into multiple components. Moreover, we designed and implemented a collaborative web browsing system in which users can search and browse their target information by discussing and watching partial pages displayed on multiple devices.

1. Introduction

Since handheld devices, such as cellular phones and PDAs, have functionality constraints on screen size, computational power, operationality, and other areas, mobile users have a difficult time browsing web pages designed for desktop PCs on their devices. In a mobile environment, multiple mobile users often move around together and simultaneously search for certain information, e.g., restaurants to go to, information on goods they want to buy, information on topics from conversation. A possible solution to this problem is to use the handheld devices of multiple users together to browse web pages. This also can improve the efficiency of collaborative search by multiple users.

We propose a new concept of *collaborative browsing*, where mobile users collaboratively browse web pages using multiple handheld devices. In collaborative browsing, a web page which consists of multiple components is divided into multiple partial pages, and each of the partial pages is distributed to a different handheld device. Then, the users can collaboratively browse the web page by watching the distributed partial pages individually or together. Since the size of a partial web page allocated to each device is not large, users can browse the web page easily and search for information more quickly. In mobile computing environments, the number of handheld devices, their capabilities, and other conditions can vary widely amongst mobile users who want to browse content. Moreover, if some components correlate with each other, they should be allocated to the handheld device of a single user for easy browsing. Therefore, in collaborative browsing, a page must be flexibly divided according to these conditions.

We designed and implemented a collaborative web browsing system. In this system, a web page is segmented into multiple components according to the structure of the page and converted into a tree whose leaves correspond to different components. This page partitioning method divides the tree into an arbitrary number of partial web pages based on given conditions. These partial pages are delivered to handheld devices. The users browse the partial web pages and search for information by watching each other's displays and talking with each other. Users should be able to find their target information in a shorter time because they can search in parallel. However, grasping the structure of the original page is difficult for users because it is divided into multiple partial pages. Thus, the collaborative web browsing system provides several overviews of the original page.

2. Collaborative browsing

First, we explain the methods of collaborative browsing. Multiple users collaboratively browsing the web are shown in Fig. 1. In collaborative browsing, users search for information by talking with each other and watching each other's displays. Moreover, users can browse web pages more easily by changing the method of browsing according to the characteristics of the pages.

- **Pages consisting of components without correlation**
  - Suppose a web page consists of components with-

out correlation, such as the top page of a portal site with various components; a site directory, news, search form, etc. For such a page, users can search for information in parallel. In this case, the user can share information verbally or make the component be displayed on their own screen using the overview function. We will explain the detailed functions of the overview in a later section.

- **Pages consisting of components with correlation** - Suppose a web page consists of components with correlation, such as a page for a restaurant with various components; the appearance of the restaurant, the location, etc. For such a page, the method of browsing depends on the goal of the user. If users search for specific information on the page, users can browse in the same way as for the top page of a portal site. However, if users want to choose a restaurant, each user can browse a web page individually and after finishing viewing their own pages, can view other screens and discuss them.

- **Pages consisting of components in a specific order** - Suppose a web page consists of components in a specific order, such as news stories or diary entries with many paragraphs and illustrations. For such a page, users can see the overview and view the components in order.

3. Problem definition

To enable collaborative browsing, a web page that consists of a number of components must be divided into multiple sets of components according to the number of handheld devices. Each set of components is allocated to a particular device as a partial page. In collaborative browsing, a web page must be flexibly divided according to the following conditions:

- **Number of devices (Number of partial pages)** - Since the number of participating handheld devices varies depending on the situation, a web page must be divided into a number of partial pages that matches the number of devices.

- **Capacity of each device** - Some devices have limited memory to process allocated pages. For example, since most cellular phones have limitations on what page size they can receive, the size of a partial page allocated to each device should meet these limitations.

- **Performance of each device** - The different performances of handheld devices should be taken into account for page partitioning. For example, if handheld devices have different screen sizes, the volume of partial pages allocated to these devices should be determined according to the ratios of the size of their screens.

Figure 1. Appearance of collaborative web browsing.

- **Functions of each device** - Since handheld devices have different functions, i.e., those to play video and audio content, components that require a special function to be viewed must be allocated to devices with that function.

- **Correlation among components** - Users can browse content easily if their handheld devices are allocated a partial page with components which correlate with each other. For example, a picture and its associated text should be allocated to the same device.

- **User preferences** - Devices should be allocated partial pages that match the preferences of users.

4. Collaborative Web browsing system

We designed and implemented a collaborative web browsing system which allocates partial pages to handheld devices based on the conditions for collaborative browsing mentioned above. Although the screens of handheld devices

are too small to display normal web pages, these pages usually consist of various components such as news, web site directories, and search forms, e.g., an index page of a portal site, and the size of such components is much smaller than that of a whole page. These components are suitable for display on handheld devices. In our system, users can change between two modes on the browser. One is the *browsing mode*, in which users view the details of a page by watching components serially. The other is the *overview mode*, in which users see the outline of the page by viewing an overview.

Our system consists of a server and handheld device clients. In this system, when users log into the server by specifying their own names and a specific keyword, they can start collaborative browsing. During collaborative browsing, other users can join by specifying the keyword. If users join or leave, the page can be re-distributed. When a user wants to join a collaborative browsing session, the user must manually send the registration to the server. The server recognizes the participation of a client in a session by receiving keep-alive messages from the client. Thus, if the server does not receive a keep-alive message from the client for a certain period, the server judges that the client has left the session. The server runs a process to convert a web page to a tree, and a process to divide the tree based on the conditions for collaborative browsing. The client runs an application to display a received partial page and to manage the clients' events. If a client sends a URL of a web page to the server, the page is divided by the server, and the divided partial pages are delivered to clients. More specifically, the server segments a web page requested by a user into components according to the structure of the web page and converts it into a tree whose leaves correspond to the components. After that, the server converts the tree into a graph and applies a graph partitioning algorithm to divide the page. By solving the graph partitioning problem according to the conditions mentioned above, the page is divided onto a number of handheld devices. For this, clients send the following information to the server in advance:

- **Capacity** - Maximum size of a web page that a device can display. The system does not deliver partial pages to clients beyond their capacity.
- **Screen size** - Dimensions of the screen of a device in pixels. The system determines the size of partial pages delivered to clients according to the ratios of their screen sizes.
- **Functions** - Specific functions of client devices, e.g., display of PNG pictures. The system does not deliver a component to a client that does not have functions to execute it.
- **Keywords** - Keywords a user is interested in. These are specified by the user beforehand as a user profile. The system delivers components with keywords to clients that specify them.

In our system, user operations, *selecting a hyperlink*, *back*, and *forward*, are synchronized over multiple clients. If a user selects a hyperlink in a web page, the target page of the hyperlink is divided and the partial pages are displayed on multiple clients.

Now, we explain the behavior of the system when a user selects a hyperlink.

1. If a user selects a hyperlink, the client sends the URL corresponding to the hyperlink to the server.
2. The server segments a web page corresponding to the URL into components.
3. The server determines the component allocation to clients according to the various conditions for collaborative browsing by solving the graph partitioning problem.
4. Each client receives the page partitioning result, which includes the identifier of the web page, its allocated components, data that specifies the layout of the components, and the information on the component allocation to all the clients. The layout data is created by eliminating all components from the HTML source file of the web page.
5. A dialog box that asks whether the user will display the received partial page (components) is shown on the screen of each client. If the user accepts, the client displays the partial page created from the received result.

Next, we explain the details of the page segmentation and partitioning methods and the client system.

5. Page segmentation

In our system, the server segments a page into components and then converts into a tree. Since there are many methods for segmenting web pages [3][5][14], our system uses a simple, widely used segmentation method. In our method, first, the entire page is set to the root node of the tree. After that, each partial page corresponding to a leaf node is divided into multiple partial pages by a widely used segmentation method, and these pages are set to child nodes of the node, as shown in Fig. 2. This procedure is repeated as long as the size of each leaf node is larger than a threshold value. The threshold value is half of the minimum capacity size of handheld devices. Here, the size of a node denotes

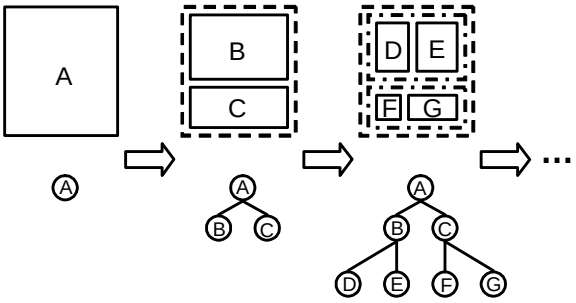


Figure 2. Page segmentation process.

the size in bytes of the HTML document of the partial page corresponding to the node. If the partial page includes images, the size of the images is added to the size of the node. Finally, the tree is simplified, e.g., an internal node with only one child node is deleted.

In a tree created by this process, the closer the components are located in the HTML page, the shorter the path length between these components (leaf nodes). For example, in the rightmost tree in Fig. 2, the path length between nodes D and E, which have the same parent node B, is 2, where the length of every edge is 1. The path length between nodes D and G, which have different parent nodes, is 4. We define this path length as the correlation between the nodes. By doing so, components located closer to each other, such as paragraphs in the same document, can be allocated to the same client.

When the process of converting a page into a tree is completed, metadata is added to the leaf nodes. The metadata is described in Ref. [10] and contains the following information:

- **Title of the component** - In this system, the title is set to the first (non-meaningless) words without tags in the component or *alt* attribute of `<img>`.

- **Size of the component** - This is the size in bytes of the partial HTML document corresponding to the component. We use the size in bytes instead of the area on the display, due to the computing requirements.

- **Functions for executing the component** - If the component has images or sounds that need special functions to execute, this information is recorded as metadata.

Using the tree and metadata, our system divides pages according to the conditions for collaborative browsing.

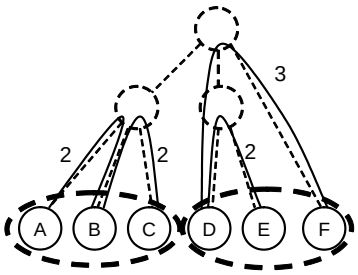


Figure 3. Example of page partitioning.

6. Page partitioning

Our group previously presented an overview of page partitioning [10]. Here, we explain a page partitioning method for dividing a page of the tree created by the page segmentation in detail. Using this partitioning method, the tree is converted into a complete graph, and then, for the converted complete graph, the *graph partitioning problem* is solved. The graph partitioning problem is used in various areas, such as VLSI circuits and distributed programs, and is NP-complete. To solve the graph partitioning problem heuristically, the *KL algorithm* [9] and the *FM algorithm* [6] have been proposed. Our proposed partitioning method extends the FM algorithm to consider the conditions for collaborative browsing and partitions a complete graph into multiple sub-graphs.

6.1. Definitions

The proposed partitioning method partitions a complete graph, which consists of leaf nodes in a tree. The weight of a node refers to the size of the corresponding component. The weight of an edge between two nodes in the graph refers to the weight of the path between the two leaf nodes in the tree. By this conversion, the weight of the edge between two nodes in the graph represents the strength of the correlation between them. An example of page partitioning is shown in Fig. 3. In this example, the page is divided into two partial pages that include the same number of components, and components in each of the partial pages correlated strongly to each other. A solid curve that connects two leaf nodes denotes the edge between the two nodes in the complete graph, however, only edges between two nodes that belong to the same partial page are shown. In this way, a page is divided into clusters of leaf nodes to minimize the sum of the weights of edges in the partitioned graphs.

Since the proposed partitioning method takes into account the conditions for collaborative browsing, this partitioning is different from conventional graph partitioning. Specifically, the proposed method introduces an objective

function, f, to evaluate the graph partitioning. This function is expressed by the following equation, where the smaller the value, the better the partitioning:

$$f = f_1 \cdot (c_2 f_2 + c_3 f_3 + c_4 f_4).$$

Here, c_2, c_3, and c_4 are weighting factors for f_2, f_3, and f_4. Using these factors, the degree of importance of each sub-function can be changed.

The function f_1 represents how many components cannot be executed on the devices that exist on the divided partial pages, and is expressed by the following equation:

$$f_1 = \frac{W_s + W_f + 1}{W_a}.$$

W_a = $\sum_{i=0}^{n-1} v_i$: the sum of the size of all objects.
n = the number of nodes.
v_i = the size of node i.
W_s = the sum of the size of nodes allocated to devices beyond their capacity.
W_f = the sum of the size of nodes allocated to devices that do not have the functions to execute them.

The function f_2 represents how far the rate of volume of partial pages allocated to the devices is from the rate of performance of the devices, i.e., the screen size, and is expressed by the following equation:

$$f_2 = \frac{1}{k} \sum_{i=0}^{k-1} \frac{(w_i - W_a \alpha_i)^2}{(W_a \alpha_i)^2}.$$

k = the number of devices.
w_i = the size of executable nodes allocated to device i.
α_i = the rate of the performance of device i to the sum of the performance of all devices.

The function f_3 represents how many nodes are not allocated to the devices contrary to the user specifications of keywords in the divided partial pages, and is expressed by the following equation:

$$f_3 = \frac{N_c}{N_k}.$$

N_c = the number of nodes not allocated contrary to the user specifications of keywords.

```
Convert Tree into Complete Graph
Create Initial k-way Partitioning
Compute f'
minf = f'
/*k-way FM Algorithm*/
Do While minf updates
    Unmark all nodes
    Do While Unmarked nodes exist
        Compute f in Unmarked nodes
        Find node_n and group_m that minimize f
        Move node_n into group_m and mark
    end Do
    Find groups that minimize f in the above loop
     and this minimum f is set to tmpminf
    If(minf > tmpminf)
        minf = tmpminf
    end If
end Do
```

Figure 4. Page partitioning algorithm.

N_k = the number of nodes that match the keywords specified by the users.

The function f_4 represents how low the correlation among nodes allocated to the same devices is in the divided partial pages, and is expressed by the following equation:

$$f_4 = \frac{\sum_{i=0}^{k-1} E_i}{E_a}.$$

E_i = the sum of weights of all edges in the minimum spanning tree that consists of nodes allocated to device i.
E_a = the sum of the weights of all edges in the complete graph converted from the tree.

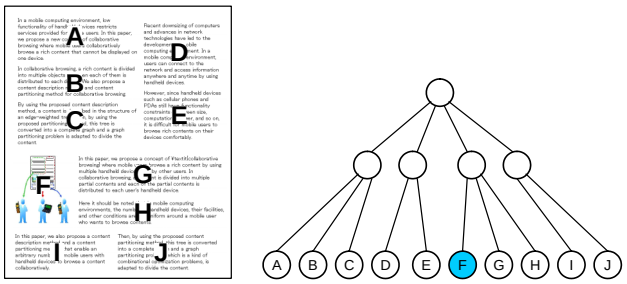
Here, the smaller f_1 is, the larger the volume of contents the users can browse. Since the objective function f represents the goodness of page partitioning, f is proportional to the volume of contents that users can browse. Therefore, we design f_1 to be multiplied by other functions in objective function f. Moreover, since f_2, f_3, and f_4 are independent of each other, these functions are added together.

6.2. Page partitioning procedure

Our algorithm is shown in Fig. 4. Next, we explain the algorithm.

[Create Initial k-way Partitioning]

The graph is initially partitioned into k groups by two procedures. First, a minimum spanning tree is found from



(a) (b)

Figure 5. Example of a web page.

Table 1. Conditions of devices.

Device	Capacity	Rate of performance	Keyword match
0	-	2	B
1	-	2	-
2	20KB	1	-

the complete graph. Second, nodes are divided into k groups by cutting $k-1$ edges from the minimum spanning tree. Here, edges are cut in descending order of weight. Finally, the set of nodes, $group_i$ $(0 \leq i \leq k-1)$, allocated to device i is output.

[k-way FM Algorithm]

In this algorithm, more suitable allocation is searched by moving nodes among groups of partial pages in the initial partitioning.

First, the value of f is calculated when a node, $node_n$, is moved to a group, $group_m$, to which the node does not belong ($node_n \notin group_m$). This calculation is performed for every node that has not yet been marked. Then, the moving operation that gives the minimum f is selected and applied to the groups. At the same time, the node that is moved to another group is marked. This operation is repeated until all nodes are marked. Consequently, among all group allocations obtained during the successive moving operations, the allocation that gives the minimum f becomes the tentative best partitioning. Second, all nodes are unmarked, and better partitioning is further searched by moving nodes among the groups. This is repeated until the minimum f is no longer updated.

6.3. Example of partitioning

An example of a web page divided into three partial contents for three devices using the proposed page partitioning algorithm is shown in Fig. 5(a). Assume the web page has the tree structure shown in Fig. 5(b), where ten leaf nodes, A to J, correspond to components A to J on the web page. Since components A to C, D to E, F to H, and I to J are located close to each other, they are classified by four parent nodes. Also assume that component F, which includes an image, can be displayed on only device 1, and the size of all components is 10 KB. Other partitioning conditions are shown in Table 1. The allocation results are shown in Table 2, where $c_2 = c_3 = c_4 = 1$ (result 1) and $c_2 = c_3 = 1$, $c_4 = 5$ (result 2). From these results, object A is allocated to device 1 in result 1 and to device 0 in result 2, because the setting $c_4 = 5$ gives high priority to correlation between objects. In this case, objects B and C, which are correlated to A, are allocated to the same device.

Table 2. Allocation results.

Device	Result 1 $c_2 = c_3 = c_4 = 1$	Result 2 $c_2 = c_3 = 1, c_4 = 5$
0	B,C,I,J	A,B,C,I,J
1	A,F,G,H	F,G,H
2	D,E	D,E

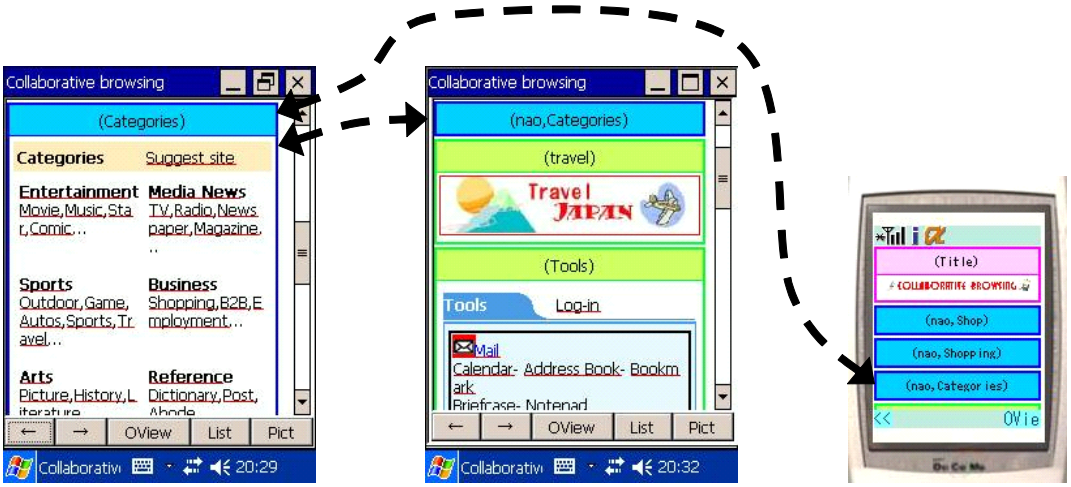
7. Client system

Each client creates a partial page from the components allocated to the client, the information on the component allocation to all clients, and the layout data the server created. This is done by embedding the allocated components and the information on the component allocation to the layout data. Here, in the browsing mode, table tags in the layout data are eliminated because they are not suitable for handheld devices with small width screens.

Screens of handheld devices (two PDAs and a cellular phone) are shown in Fig. 6, where three users are in browsing mode. In browsing mode, each of the allocated components is located in a rectangle to make clear the border of the component. In addition, the title of the component is specified at the top of the rectangle. Each component allocated to another user is expressed by only a rectangle with the name of the user and the title of the component. Rectangles are colored according to the users. For example, in Fig. 6, on the screen of user "kazu", the rectangle of the component labeled "Categories" allocated to user "nao" and the rectangles labeled "travel" and "Tools" allocated to "kazu" are displayed. If user "kazu" wants to view the component labeled "Categories", he can ask "nao" to show the screen of his device to "kazu". With this interface, users can easily share the information orally. This is the most important element in collaborative browsing.

In browsing mode, users can perform the following operations:

- **Selecting a hyperlink** - When a user selects a link, the



User: nao User: kazu User: taku

Figure 6. Screens in the browsing mode.

corresponding web page is partitioned and delivered to various users. The user who selects the link can determine the users to whom the page is delivered. With this function, users can split up web pages to search for information in parallel.

- **Moving back and forward** - These are similar to the back and forward operations of conventional web browsers. However, a user who performs this operation can select whether the same operation is performed for others simultaneously. If a user receives a message that another user has performed a back or forward operation, a dialog box that asks whether the corresponding page should be displayed is immediately shown.

- **Switching to overview mode** - This operation displays the overview of the page that is currently displayed.

- **Showing an event list** - This operation shows a list of events and actions performed by all users, such as receipt of a partitioning result and use of the back operation. Users can re-execute actions by selecting them from the list.

In the system, when a user selects a hyperlink, the user can select other users who receive the partitioning result of the hyperlinked page. So that a user can separate a web browsing session by using this function. Also, a user can perform an operation to integrate browsing sessions when the user selects a hyperlink. Specifically, when a user selects a hyperlink, a dialog box is displayed on the other users' screens to ask whether they wish to follow (join) the session.

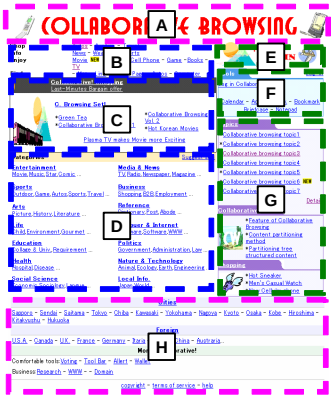
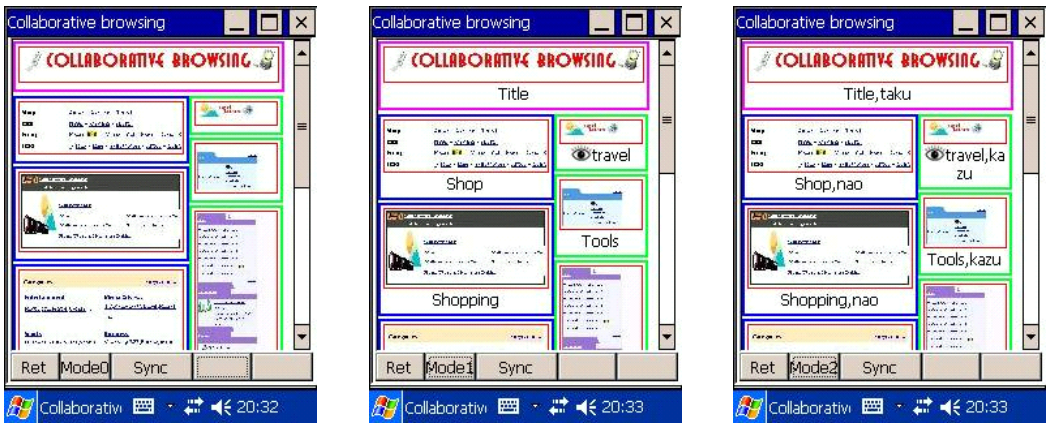


Figure 7. Web page allocation.

The original web page of the partial pages is shown in Fig. 6, and the allocation of components to users is shown in Fig. 7. This example page is made of the top page of an actual portal site by only replacing text and images. The web page is segmented into eight components from A to H. Components B to D, E to G, and A and H are allocated to users "nao", "kazu", and "taku", respectively. The total size of components allocated to "nao", "kazu", and "taku" are about 24, 32, and 11 KB, respectively. From these results, components that are located close to each other are allocated to the same device. Also, the cellular phone is allocated fewer components than the PDAs.

In our system, graphical overviews are provided, as shown in Fig. 8. In Fig. 8(a), images represent components allocated to the users corresponding to the colors of the rectangles. These images are created by the server as part of a



(a) (b) (c)

Figure 8. Screen in the overview mode.

snapshot of the entire web page, and each image is a hyperlink. If a user selects a hyperlink corresponding to a component allocated to him, the mode switches to the browsing mode, and the component is displayed on the screen. If a user selects a hyperlink corresponding to a component allocated to another user, a dialog box is displayed on the screen of the recipient of the component that asks whether the component should be displayed immediately. Users can change the degree of detail of the overview. In Fig. 8(b), titles of components are displayed below the corresponding images, and an eye icon, which indicates that a component is currently being viewed by the user in browsing mode, is displayed next to the title "travel". In Fig. 8(c), names of owners of the components are also displayed. Here, the overview keeps the page layout of the original web page when the degree of detail is low but does not when the degree of the detail is high.

With browsing mode, overview mode, and their functions, users can exchange information, talk, and watch each other's displays.

Client systems were implemented on cellular phones and PDAs using the Java language. We used NTT DoCoMo SH900i, F900i, and iPAQ Pocket PC h5500. The capacity of the cellular phones was set to 20 KB, while the PDAs had no limitation. The screen resolution of the PDAs was about 1.2 times larger than that of the cellular phones. The server system was implemented on Red Hat Linux using the Java language and PHP (hypertext preprocessor).

8. Evaluation

We show results of our performance evaluation to verify the effectiveness of our implemented system. In this evaluation, the weighting factors in the objective function f were set to $c_2 = c_3 = c_4 = 1$, and keywords were not set to specify user preferences.

8.1. Experiments

8.1.1 Experiment 1

In this experiment, we compared our system with the standard WWW browsers of cellular phones and PDAs. Since most web pages designed for desktop PCs exceed the capacity of cellular phone web browsers, we constructed a restaurant search site for the evaluation by simplifying and using a specific site. Note that our system can browse web pages designed for desktop PCs because it takes the capacity of mobile devices into account.

In our developed site, a user first selects a hyperlink that specifies a type of restaurant on the top page, which includes hyperlinks for eight types. Then, the page corresponding to the selected link is displayed, including links to restaurants of the selected type. The user can display detailed information about a restaurant by selecting a corresponding link on the page for each type. This site had web pages for thirty eight restaurants. The size of the top page was 42 KB, including images. The size of the page that included links to restaurants of a specific type was about 70 KB. The size of the page that included detailed information about a restaurant was about 30 KB. Using this site, we performed the following experiment.

Eighteen subjects from 22 to 26 years old were tested in six different conditions, which are described below. These subjects had never used our system. In conditions 1 to 5, six groups of three subjects searched for a restaurant of a particular type, where the target restaurant and type were randomly chosen for each group. For conditions 2 to 5, the order of the four tests (Conditions 2, 3, 4, and 5) was

randomly determined for each group, and each group was given five minutes to learn how to use our system before starting the first test. In condition 6, four groups of four subjects took the same test. On the restaurant search site, users can display target restaurants by following two links.

- **Condition 1** - Three cellular phones with standard WWW browsers.
- **Condition 2** - Three cellular phones equipped with our system.
- **Condition 3** - Two cellular phones and one PDA equipped with our system.
- **Condition 4** - One cellular phone and two PDAs equipped with our system.
- **Condition 5** - Three PDAs equipped with our system.
- **Condition 6** - Two cellular phones and two PDAs equipped with our system.

Generally, the time necessary to find information depends on the structure of web sites and the purpose of browsing. This experiment represented a situation in which subjects search for specific information that they know exists but do not know its precise location on a web site of the tree structure whose root is the index page. More specifically, it represented a situation where users want to know detailed information about a restaurant, but they know only the name and type.

8.1.2 Experiment 2

The same subjects as in experiment 1 were tested in three different conditions, shown below. In all conditions, six groups of three subjects searched for a component with a particular name, where the target component was chosen randomly. Here, new 17 KB pages with images were prepared for this experiment and were segmented into twelve components, each including a long piece of text.

- **Condition 1** - Three cellular phones with standard WWW browsers, where table tags in HTML texts were eliminated.
- **Condition 2** - Three cellular phones equipped with our system, where components are searched for in browsing mode.
- **Condition 3** - Three cellular phones equipped with our system, where components are searched for in overview mode.

8.1.3 Experimental results

The average search time for each condition of experiment 1 is shown in Table 3. The average search time for condition 1 is about 40 seconds longer than that in condition 2. We confirmed a significant difference between the two results by using Welch's t-test ($p < 0.05$) [13]. This shows the effectiveness of our system when multiple users collaboratively search for information. In condition 1, users were often in trouble with teaching each other the locations of components because users viewed different pages. In fact, users made mistakes in selecting links on twelve occasions. This shows that users cannot grasp the structure of pages in which table tags have been eliminated. In conditions 2, 3, 4, and 5, the more PDAs used, the shorter the search time was. However, we cannot confirm significant differences between conditions 2 and 3, 3 and 4, and 4 and 5. If results where users made obvious mistakes are removed, we can confirm a significant difference between conditions 2 and 3. This shows that using PDAs shortens the search time compared with using only cellular phones. We cannot confirm significant differences between other conditions because the variance between groups was very large. However, by examining the search times of each group in more detail, we determined that, the more PDAs used, the shorter the search time was in every group. This is because a PDA can easily scroll through pages using the scroll bar and the stylus, i.e., the functionality of a PDA is better than that of a cellular phone. This was also confirmed for condition 6, where the average search time was shorter than that for condition 4 and longer than that for condition 5. From these results, the ratio of allocated component sizes should be determined according to the functionality of client devices.

For condition 3, one group made a mistake in selecting links once. However, this happened less frequently than for condition 1. Since the users were given a task to find information on a specific restaurant, users did not split up to search for the information.

We performed an additional experiment where some groups of three subjects collaboratively searched for information using three PDAs with a standard WWW browser. In a group, after a subject found the web page that included the target information, the subject taught the others how to get to the page. While the subject completed the task in a shorter time than that when using the collaborative web browsing, it took much time to teach the others. As a result, the average search time of the subjects in the group was longer than that when using the collaborative web browsing in Condition 1. In another group, some subjects waited for the others to find a hyperlink that they had already found, so that all the subjects reached the target information at the same time by teaching each other and displaying the same web pages on their screens. As a result, it

Table 3. Result of experiment 1.

Condition	1	2	3	4	5	6
Time(sec)	100.3	60.9	58.1	48.4	41.2	47.5

Table 4. Result of experiment 2.

Condition	1	2	3
Time(sec)	58.9	45.6	28.2

took longer to search than when using the collaborative web browsing due to the waiting time. From these results, it is shown that while devices with a large display and a standard WWW browser achieve short search times to complete simple tasks even by a single user, original browsers don't have enough functions for multiple users to collaboratively complete tasks.

The average search time for each condition of experiment 2 is shown in Table 4. The average search time for condition 1 was longer than that for conditions 2 and 3. We only confirmed a significant difference between conditions 1 and 3 but not between 1 and 2. This is because users made mistakes in selecting links seven times for condition 1, and thus, the variance was very large. The search time for condition 3 was much shorter than that for condition 2. We confirmed a significant difference between these conditions. This shows the effectiveness of the overview function of our implemented system. Users can find information in a shorter time because they rarely scroll across screens and can find out easily which user is viewing a specific component.

In this experiment, we found interesting user interactions. In collaborative browsing, users have a difficult time showing their displays to all other users simultaneously or showing them to other users who are more than 1 meter away. To solve these problems, some subjects read out information from their allocated partial pages and others exchanged devices. These interactions are useful for collaborative web browsing. Specifically, conversation is effective for getting a consensus. Actually, in this experiment, groups in which subjects rarely spoke with each other frequently made mistakes, e.g., two or more subjects often selected links at the same time. To avoid such mistakes, a mechanism to accept only the first operation and ignore later operations may be effective.

8.2. Questionnaire

We also provided a questionnaire to the eighteen subjects on the functionality of the system. About 67% of the subjects thought that our system made it easier to search a component than standard web browsers. In addition, about 70% of the subjects thought that web pages displayed on devices in browsing mode were easy to view and share with each other. This shows that users can easily get information on the names of objects allocated to other users from the screen in browsing and overview modes. In collaborative browsing, the ease with which users can share display is very important because there are many such situations. For example, when users want to choose a restaurant, they usually watch multiple screens that display the information about different restaurants and talk to each other.

Many subjects said they enjoyed collaborative browsing very much. Actually, during the experiments, we saw some groups laugh while browsing contents. From these facts, we believe collaborative browsing is useful not only for efficient browsing but also amusement.

In collaborative web browsing, web page partitioning is performed so that components that are close together in a web page are allocated to the same device. Thus, in most cases, allocated components are easy to browse for users. However, some users may be allocated useless or irrelevant contents that are not directly related to the main content of the web page or the target information for the browsing. Actually, such cases sometimes happened in the experiment and subjects who were not allocated the important contents told the others the fact and/or started to watch a nearby user's screen. From the results of the questionnaire, it is shown that such subjects did not feel uncomfortable. This seems to be due to the effect of collaborative browsing that subjects in the same group feel togetherness.

8.3. System performance

We performed an experiment to evaluate the performance of the page segmentation and page partitioning algorithms written in Java on a PC with a 1.5 GHz CPU and 1 GB of main memory. We selected fifty typical web pages from popular web sites, such as Yahoo.com, About.com, and MSN.com, and measured the time to segment each page into components and to partition the page when two PDAs and one cellular phone were used. The average time for page segmentation was 3.686 seconds. The average time to retrieve the size information of images included in a page was 2.689 seconds because most web servers that manage these pages are far from our PC on the Internet. A possible solution to this problem is to estimate the image size from the *height* and *width* attributes in the *img* tag. Moreover, since we did not use a DOM tree commonly used in page analysis, but instead used conventional text analysis to tolerate HTML syntax errors, segmentation took a long time. The average time for page partitioning was 0.226 seconds.

These times are much shorter than that necessary to download and browse web pages.

9. Related work

To browse rich contents using multiple devices, Ubicomp Browser [1], situated computing [12], and task computing [11] provide architectures in which mobile users can use devices set around them, such as large displays and speakers, in pervasive computing environments. In these works, each content is allocated to specific device according to the type of the content. For example, a sound content is allocated to the speaker and a movie contents is allocated to the display. Our approach is completely different to these approaches because our system dynamically divides contents considering conditions for collaborative web browsing. WebSplitter [7] also uses devices around users and allocates different versions of pages to them. This is done by filtering the original pages with specific metadata in advance according to the performance of devices. Our system is different from this system because web pages are divided dynamically.

Some studies have been done in which multiple users can browse web pages collaboratively using a normal or extended WWW browser. CoWeb [8] provides functions to enable users to fill HTML forms collaboratively, to highlight special parts of the content, to communicate with each other by typing, and so on. The system in [2] creates such functions by a proxy-based approach. In [4], the authors discuss requirements for web interaction among multiple clients. The collaborative browsing method proposed in this paper aims to browse detailed web pages using simple handheld devices, and thus, it is different from these studies.

10. Conclusion

We proposed a new concept of collaborative browsing and a page partitioning method. We also designed and implemented a collaborative web browsing system. In our system, users can collaboratively browse web pages by watching multiple screens and talking with each other. To do so, the client system provides several functions for collaborative activities.

For future work, we plan to deal with pages that use Cookies and scripts. In addition, we are working on allowing a user to browse a page not only with friends but also with the general public when friends have no devices with the functions necessary to execute some components of the page. To do this, a mechanism to call for participants who have devices with the required functions is needed. In our system, we use only the screen size as the metric of device performance. We plan to use the communication, computing, and imaging speeds as metrics. We also plan to verify the effectiveness of our system and observe users' interactions when they discuss the browsing strategy such as to decide which page to select.

Acknowledgements

This research was partially supported by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment" and Grant-in-Aid for Young Scientists (A)(16680005) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] M. Beigl, A. Schmidt, M. Lauff, and H. Gellersen. The ubicompbrowser. In *4th ERCIM Workshop on User Interfaces for All*, Oct. 1998.

[2] G. Cabri, L. Leonardi, R. Gimson, and T. Wiley. Supporting cooperative www browsing: a proxy-based approach. In *Euromicro Workshop on Parallel and Distributed Processing*, pages 138–145, Feb. 1999.

[3] Y. Chen, W. Ma, and H. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *WWW'03*, pages 225–233, May 2003.

[4] A. Coles, E. Deliot, and T. Melamed. A framework for coordinated multi-modal browsing with multiple clients. In *WWW'03*, pages 718–726, May 2003.

[5] D. Embley, Y. Jiang, and Y.-K. Ng. Record-boundary discovery in web documents. In *ACM SIGMOD'99*, pages 467–478, May/June 1999.

[6] C. Fiduccia and R. Mattheyes. A linear time heuristic for improving network partitioning. In *Design Automation Conference*, pages 175–181, June 1982.

[7] R. Han, V. Perret, and M. Naghshineh. Websplitter: A unified xml framework for multi-device collaborative web browsing. In *ACM Conference on Computer Supported Cooperative Work 2000 (CSCW 2000)*, pages 221–230, Dec. 2000.

[8] S. Jacobs, M. Gebhardt, S. Kethers, and W. Rzasa. Filling html forms simultaneously: Coweb - architecture and functionality. In *WWW'96*, pages 1385–1395, Dec. 1996.

[9] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2):291–307, Feb. 1970.

[10] T. Maekawa, T. Uemukai, T. Hara, and S. Nishio. Content description and partitioning methods for collaborative browsing by multiple mobile users. In *Int'l Workshop on Mobility in Databases and Distributed Systems (MDDS 2005)*, pages 1068–1072, Aug. 2005.

[11] R. Masuoka, B. Parsia, and Y. Labrou. Task computing-the semantic web meets pervasive computing. In *International Semantic Web Conference (ISWC'03)*, pages 866–881, Oct. 2003.

[12] T. Pham, G. Schneider, and S. Goose. A situated computing framework for mobile and ubiquitous multimedia access using small screen and composite devices. In *ACM Multimedia 2000*, pages 323–331, Oct./Nov. 2000.

[13] B. Welch. The generalization of student's problem when several different population variances are involved. *Biometrika*, pages 28–35, 1947.

[14] Y. Yang and H. Zhang. Html page analysis based on visual cues. In *6th International Conference on Document Analysis and Recognition*, pages 10–13, Sept. 2001.