

The Mobile Web-Com e-Learning System for Mobile Phones

Susumu Shibusawa[†], Tatsuhiro Yonekura[†], Michitoshi Niibori^{††}, and Munehisa Kamata^{†††}

[†]Department of Computer and Information Sciences, Ibaraki University, Hitachi, Ibaraki 316-8511, Japan

^{††}Learning-i Co. Ltd., Hitachi, Ibaraki 316-8511, Japan

^{†††}Hitachi Business Solution Co. Ltd., Yokohama, Kanagawa 231-0062, Japan

Summary

Recent growth in network environments has led to a surge in demand for Internet-based e-learning in schools and businesses. To meet this demand, the Web-Com remote education software has been developed with voice and annotation management functions using multiple layers. Meanwhile, with the growing popularity of mobile phones, there is expected to be growth in remote education systems using mobile phones that have superior portability and accessibility. In this study, therefore, Web-Com content for PCs is adapted to design a Mobile Web-Com e-learning system that runs on mobile phones, and this system is tested by running it on simulators and on actual mobile phone equipment. This system operates in a BREW environment, and allows PC Web-Com annotations and voice-over content to be played back on mobile phones.

Key words:

e-Learning, Mobile Phone, Annotation, Content Transformation, BREW.

1. Introduction

Recent growth in network environments has led to a surge in the demand for Internet-based e-learning in schools and businesses, and in particular there has been growth in Web-based training (WBT) via the Internet [1], [2]. E-learning can be broadly classified into synchronous and asynchronous learning models. In synchronous models, the teacher interacts with the students in real time, whereas in asynchronous models, the teacher and students do not have to be present at the same time. In e-learning, the teacher and students are often in different locations, which can make it harder for the students to keep themselves motivated and can result in poorer communication between the teacher and students. Furthermore, it is hoped that e-learning systems can be improved by allowing classes to be implemented more efficiently and by automating the administration of grades.

As e-learning systems become more sophisticated, they will be required to implement advanced content creation functions, and there will be a demand for systems that can be used easily even by teachers who are unfamiliar with such systems. User-friendly annotation functions are indispensable tools that help when teachers produce content and when students take part in learning, and

systems that implement the annotation of web content have already been reported [3], [4], [5]. The addition of annotations to web content allows memos to be written and used with existing content, and can thus enrich the handling of content.

To improve the development of teaching materials by teachers and the ease of learning by students, the “Web-Com” system has been developed as a web-based interactive browser for enhanced teaching [6], [7]. Web-Com allows annotations and voice recordings to be added to web content, and uses multiple layers to manage these annotations. This system can also reproduce the processes whereby annotations are added from the operation history.

On the other hand, with the spread of portable terminals and the improving performance of these terminals, various kinds of content are being developed for portable terminals. In e-learning for PCs, the places where learning can take place are limited, but in mobile learning (m-learning) it can be used at any time even when away from home or while traveling, allowing time to be used more effectively for learning [8], [9], [10], [11], [12], [13]. M-learning is characterized by mobility and accessibility, and can engender collaboration between users depending on how the classes are structured [8].

However, since PCs and mobile phones differ with regard to features such as their display resolution, character size, content display magnification and processing capabilities, content produced for ordinary PCs will not display correctly on mobile phones [14]. In particular, in content consisting of multiple items of content, the mutual relationships between each part of the content changes between PCs and mobile phones so that not only is the content displayed incorrectly, but sometimes the content is changed to an incorrect form. The same applies to Web-Com content, where changes to the relationships between HTML content and annotations in particular resulted in the characteristic of PC content not being expressed on mobile phones.

Therefore in this study we developed the Mobile Web-Com e-learning system for mobile phones [15] by

converting Web-Com content for use on mobile phones. Mobile Web-Com runs in the BREW environment [16], [17], and can be used to play back PC Web-Com content with annotations and voice additions on mobile phones. Mobile Web-Com is the first system created on mobile phones according to the original concept. In this study we also confirm the operation of this system in simulations and on real mobile phones, and we evaluate its performance.

In this paper Section 2 first discusses the main functions of Web-Com and the methods used for the conversion of annotations. In Section 3 we discuss the design of Mobile Web-Com system configuration, and in Section 4 we discuss the implementation of this system. Section 5 discusses the results obtained by operating this system in practice, and finally in Section 6 we present a summary and issues for further study.

2. Preparation

2.1 Web-Com

The Web-Com remote education software provides a PC-based learning system that features functions that use multiple layers for the addition of annotations, sharing these annotations, making voice recordings, replaying the annotation history, and so on. So far, it has been used in classes such as university courses on the principles of information processing and basic calculus, and for junior high school math lessons [6], [7].

This system consists of the Web-Com software, an educational content server and an voice server. The educational content server is an FTP server used for the management of lesson content and of study records produced by Web-Com. This server allows study records to be shared asynchronously between teachers and students. The management of study records is performed using a UNIX-based file system whereby the teachers and students can be assigned detailed access privileges to their own respective study records.

Web-Com is a web browser that uses the display engine from Microsoft's Internet Explorer™. Fig. 1 shows an example of the Web-Com display. In addition to basic browsing functions, Web-Com also has various functions that support Web-based education. These functions are summarized below.

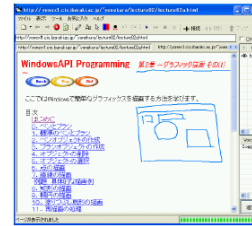


Fig. 1 Web-Com screenshot.

(1) Annotation function

This function allows annotations to be added to Web content displayed in Web-Com. This makes it possible to perform operations such as writing memos on Web-Com content in the same way as one might write a memo in an ordinary printed textbook.

(2) Annotation sharing function

Web-Com allows annotations to be shared with other users. Annotation sharing methods are provided to support both synchronous and asynchronous learning modes.

(3) Multiple layers

In the Web-Com model, a layer is a transparent sheet on which annotations can be added. Web-Com supports multiple layers which are superimposed on each other, and the added annotations are managed in multiple layers. For each layer, it is possible to specify whether or not it should be displayed and whether or not annotations can be added to it. The benefits of using multiple layers are as follows:

- Improved interaction with other users
- Simplified classification of annotations
- Simplified re-use of annotations

When a teacher holds a lesson using the multi-layer annotation sharing function, if there is a question from a student, the question can be shared between the teacher and students by using a single layer. Furthermore, by writing hints relating to a question and the correct answer in separate layers, it is not only possible to deal with questions in stages, but it is also possible to re-use these layers.

(4) Voice recording function

Web-Com is equipped with a function for making voice recordings. Ogg Vorbis [18] is used as the format of these recordings in order to keep the file sizes down.

(5) Voice interaction functions

By using the voice server, it is possible for multiple users to engage in voice interaction in real time. This function is mainly used for synchronous learning modes.

(6) Study records playback function

Web-Com is equipped with a function for playing back the annotation and voice study records in time-series fashion. The information included in the study records consists of actions such as the addition of annotations, the switching of selected layers, changes to the scroll positions of lesson content, and the recording start times and filenames of voice data. This information can be used to implement fast-forward, rewind and random seek functions, allowing users to specify any playback position.

Web-Com lessons can be recorded with a substantially smaller file size than would be achieved if lessons were recorded on video. The file size of the study records is approximately 20 MB per hour, comprising approximately 15 MB of voice data and 5 MB of annotation data.

2.2 Transformation of Annotation Files

Since mobile phones and PCs have different display widths, using a mobile phone to display PC annotations produced in Web-Com would result in a mismatch between the background HTML content and the annotation. This situation is illustrated in Fig. 2. Also, due to differences in their processing performance, a mobile phone would be unable to handle the computational load involved in processing and displaying annotations at the same temporal period as Web-Com. It is thus difficult for annotation files produced in Web-Com to be used directly on mobile phones, and instead the annotation files made for PCs have to be converted for use on mobile phones.

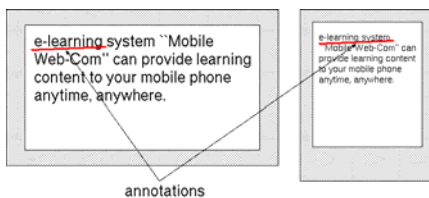


Fig. 2: An example of mismatch between an annotation and the background HTML

For a certain integer n , an annotation A is an ordered set of n line segments a_1, a_2, \dots, a_n , where A is expressed as follows:

$$A = \langle a_1, a_2, \dots, a_n \rangle \tag{1}$$

For simplicity, we will assume that the line segments of annotation A are all joined consecutively, and in cases where they are not joined consecutively, annotation A is treated as a set of consecutive parts A_1, A_2, \dots . For a consecutive annotation A and $1 \leq i \leq n$, line segment a_i is a line segment connecting two points $p_{i-1}(x_{i-1}, y_{i-1})$ and

$p_i(x_i, y_i)$. The equation for a line segment a_i is expressed as follows:

$$y - y_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(x - x_i) \tag{2}$$

where points $p_0(x_0, y_0)$ and $p_n(x_n, y_n)$ represent the start and end points of annotation A respectively.

If t_1 is the unit time taken to record one line segment of annotation A , then annotation A consists of the series of line segments connecting the coordinate points saved at each interval t_1 . The series of points p_0, p_1, \dots, p_n in annotation A consisting of n line segments is a time series recorded at each time interval t_1 , and the time needed to record annotation A is equal to nt_1 . An annotation recorded in time nt_1 is normally played back or drawn in time nt_1 , but in fast forward mode it can be played back in less time.

Next, a series B of consecutive line segments consisting of m line segments b_1, b_2, \dots, b_m where $m \leq n$ is expressed as follows:

$$B = \langle b_1, b_2, \dots, b_m \rangle \tag{3}$$

For $1 \leq i \leq m$, if line segment b_i is a line segment connecting two points $q_{i-1}(x'_{i-1}, y'_{i-1})$ and $q_i(x'_i, y'_i)$, then the equation for line segment b_i can be expressed in the same way as the equation for line segment a_i . We will assume that the unit time taken to record a single line segment of annotation B is t_2 . The series of points q_0, q_1, \dots, q_m of annotation B consisting of m line segments consists of a series of points stored at each time interval t_2 , and the time needed to record annotation B is mt_2 .

When the start point $p_0(x_0, y_0)$ of annotation A is the same as the start point $q_0(x'_0, y'_0)$ of annotation B and annotation B is an annotation corresponding to annotation A , it is assumed that annotation B is an annotation obtained by transforming annotation A . In this case, the recording times of annotations A nt_1 and B mt_2 are equal, so the unit recording time t_2 of the line segments of annotation B can be expressed as follows:

$$t_2 = \frac{n}{m}t_1 \equiv kt_1 \tag{4}$$

where $k = n/m$ is the reduction scale factor of the annotation transformation. When k is an integer, the sample series q_0, q_1, \dots, q_m of annotation B is a partial set of the sample series p_0, p_1, \dots, p_n of annotation A .

Fig. 3 shows an example of an annotation transformation where the reduction scale factor k is an integer. In this figure, the dotted line and the solid line respectively represent annotation A before the transformation and annotation B after the transformation. In Figs. 3(1) and (3), three line segments are transformed into a single line segment. In Fig. 3(2), four line segments are transformed into two line segments.

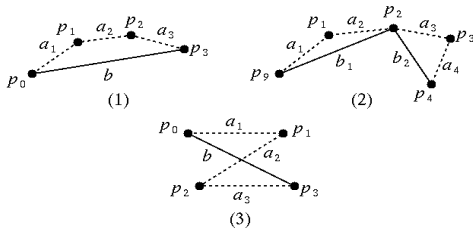


Fig. 3: Annotation transformation examples

In order to convert PC content for use on mobile phones, we need a system for converting the annotations. From the design viewpoint of an annotation conversion system, we introduce a threshold value T with regard to the unit time of the line segments in the converted annotation B . The threshold value T represents a rough target for the unit time duration t_2 of the line segments in annotation B . With respect to the annotation A with a line segment recording time of t_1 produced by Web-Com, when $kt_1 \leq T < (k+1)t_1$, the reduction scale factor of the transformed annotation B is $k = \lfloor T/t_1 \rfloor$, where the notation $\lfloor \cdot \rfloor$ signifies round-off. Here, the line segment recording time of annotation B is equal to $t_2 = \lfloor T/t_1 \rfloor t_1$.

Figs 3(1) and (3) show the cases corresponding to threshold values of $3t_1 \leq T < 4t_1, t_2 \leq T < 2t_2$, whereby the three line segments of annotation A are transformed into a single line segment in annotation B . Note that in Fig. 3(3), the characteristics of the annotation before transformation differ more significantly from the characteristics after transformation than in the case of Fig. 3(1), resulting in a high possibility of the annotation being misunderstood by the learner. In such cases, the content creator or content administrator can improve the quality of the content by resetting the threshold value of the content transformation. It may also be necessary to use a smaller transformation threshold for annotations that were recorded with very quick movements.

3. Design of System Configuration

This section focuses on the overall design of Mobile Web-Com system configuration and functions. The Mobile Web-Com system comprises the following elements:

- Mobile Web-Com player
- Content management server
- Content server

In the Mobile Web-Com system, content is auditioned by operating a Mobile Web-Com player on a mobile phone. The content management server and content server are HTTP servers situated on the Internet. It is possible (but not essential) for these two servers to be situated in the same place.

The content server is used to store content created by the content creator, and the content management server contains content lists. When content is created, the content administrator places the content on the content server and adds a link to this content to the content list. The Mobile Web-Com player retrieves the content list from the management server, and thereby retrieves the content from the content server.

3.1 System Functions

The Mobile Web-Com player has functions for playing back content that has been transformed for use in Mobile Web-Com. The functions available during playback include functions for pause, resuming, fast forwarding and rewinding the content. Also, to compensate for the small screen size, it includes an annotation jump function which scrolls the screen to the position where an annotation is written. With the current specifications, content may either be downloaded from the content server or accessed from content that is already stored in the mobile phone. The operating procedure of Mobile Web-Com is shown in Fig. 4. After the Mobile Web-Com player has retrieved the content list, it retrieves the content files according to the contents of the content list.

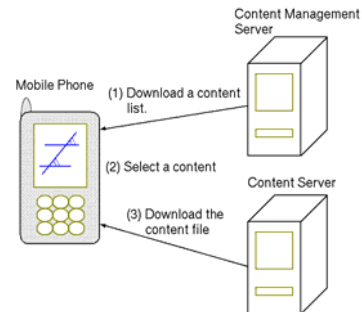


Fig. 4: Mobile Web-Com operating procedure

3.2 Mobile Web-Com Content

Mobile Web-Com content is produced by transforming PC Web-Com content files for use with Mobile Web-Com and using a content generation script to add content headers and link the files together. The file elements of the converted Mobile Web-Com content are shown below:

- Annotation files
- Background image files
- Voice files

3.3 Transformation of Content Files

In this section we describe the functions of the Mobile Web-Com player and we present methods for the transformation and creation of content. Table 1 shows the main content files produced in PC Mobile Web-Com. To produce Mobile Web-Com content files, the files shown in Table 1 are transformed into Mobile Web-Com content files according to the following procedure.

1. Use a converter to transform the annotation files.
2. Convert the HTML files into JPEG images and partition these images.
3. Transform the voice files into QCP format.
4. Use a content generation script to produce content files.

These steps are described below.

Table 1: The main types of PC Web-Com content file

Files	Contents
Annotation files	Images drawn with lines and text information
HTML files	HTML files that form the basis of the content
Voice files	Audio commentary on the content

3.3.1 Transformation of annotation files

Since PCs and mobile phones have different display resolutions, if annotations are drawn on mobile phones at the same coordinates as those saved in a PC Web-Com system, then there will be a mismatch between the background HTML image content and the annotation. Also, in a PC Web-Com system the annotations are stored at a period of approximately 20 msec, while in mobile phones the processing load required to draw annotations with such a short period would be too great. We therefore produced a converter to convert the annotation files produced in a PC Web-Com system so that they can be handled by mobile phones. Inside this converter, a temporal threshold value is applied to the annotation line segments, whereby k PC Web-Com line segments that are

below the threshold are converted into 1 mobile Web-Com line segment.

The PC Web-Com annotations are in text format, but by processing them so that they are transformed into binary format, it is possible to compress the size of the annotation files and improve the read-out speed, thereby reducing the download time. The Mobile Web-Com annotations include two types of instructions — line-drawing instructions and scroll instructions. The formats of these instructions are shown in Table 2. Also, the specifications of each element of the annotations in Mobile Web-Com are shown in Tables 3 and 4.

Table 2: Annotation instruction format

Instruction No.	Instruction type	Format
1	Line-drawing	1, x coordinate of start point, y coordinate of start point, x coordinate of end point, y coordinate of end point, color setting flag, red element, blue element, green element, α element, time
2	Scroll	2, x coordinate, y coordinate, time

Table 3: Specifications of line-drawing instruction elements

Element name	Type	Range	Size (bytes)
Instruction No.	int	1~1	1
x coordinate	int	0~480	2
y coordinate	int	0~960	2
Color setting flag	int	0~1	1
Color element	int	0~255	3
Time	int	0~1677215	3

Table 4: Specifications of scroll instruction elements

Element name	Type	Range	Size (bytes)
Instruction No.	int	2~2	1
x coordinate	int	0~480	2
y coordinate	int	0~960	2
Time	int	0~1677215	3

Table 5 shows an example of how PC Web-Com drawing instructions are transformed according to the Mobile Web-Com specifications. The results of transforming into the Mobile Web-Com format are shown in hexadecimal representation. In this example, an instruction consisting of 34 bytes in the PC Web-Com text format can be expressed in 17 bytes in the binary format of Mobile Web-Com, corresponding to a 50% reduction in the size of the drawing instructions. Table 6 shows the size of annotations in raw PC Web-Com content, and the size of the annotations after they have been transformed for Mobile Web-Com. Also, in Mobile Web-Com, since it is

unnecessary to perform processing to convert numerical values from text format when the annotations are read in, the data can be read in more quickly.

Table 5: Example of the transformation of drawing instructions

System	Instruction	Size	Format
Web-Com	LINE, 50, 100, 150, 300, 0, 0, 255, 0, 1930	34byte	text
Mobile Web-Com	01 32 00 64 00 96 00 2C 00 01 00 00 FF 8A 07 00	17byte	binary

Table 6: Size of annotation files before and after transformation

Content No.	Before transformation (Kbyte)	After transformation (Kbyte)	Threshold (msec)
1	108	22	50
		16	100
		13	150
2	56	11	50
		8	100
		7	150

3.3.2 Transformation of HTML content

Since the display resolutions of PCs and mobile phones are generally different, if the HTML content of PC Web-Com is displayed on a mobile phone, then inconsistencies will arise in the positions of line breaks in the horizontal direction of the HTML content. An example of this situation is shown in Fig. 5. It is not easy to predict where the line breaks in an HTML file will occur on every line. In this system, to avoid changes to the line break positions in the HTML file, the HTML file is converted into and displayed as an image, and a function is provided for scrolling this image. The procedure and criteria for converting HTML files for use in Mobile Web-Com are shown below.

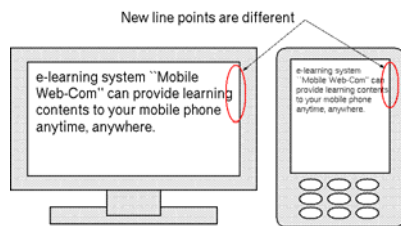


Fig. 5: Changes in line break positions caused by differences in display resolution

1. Convert HTML file into JPEG image.
2. Scale image down to 480x960 pixels or less.
3. Process the image as follows:
 - If the maximum x coordinate is 240 pixels or more, split the image horizontally into two parts.

- If the maximum y coordinate is greater than 320 pixels and less then or equal to 640 pixels, split the image vertically into two parts.
 - If the maximum y coordinate is more than 640 pixels, split the image vertically into three parts.
4. Name the partitioned image files ij .jpg, where $0 \leq i \leq 1$ and $0 \leq j \leq 2$ are the horizontal and vertical partition numbers, respectively.

Fig. 6 shows an example of an HTML image that has been partitioned in this way. Fig. 7 shows the result of connecting these images together and displaying them on a simulator.



Fig. 6: Example of a partitioned HTML image



Fig. 7: Example of images connected together on a simulator

3.3.3 Transformation of voice files

In PC Web-Com, the voice data is stored in Ogg format. For mobile phones, this is converted into Qualcomm PureVoice (QCP) format [19]. This is done by first transforming the Ogg voice files into WAV format, and then transforming the WAV files into QCP format by using PureVoice or PureVoice Converter.

3.3.4 Content generation scripts

Content generation scripts are produced in order to use files produced by the above procedure in Mobile Web-Com. A content generation script performs the following procedure.

1. Each content file is compressed with gzip.
2. A header generation program is run to generate content header files, and content headers are generated.
3. The content headers and gzip files are linked together.

The content headers contain information on the content version, the content header size, the number of stored files, and the filename and file size of each compressed file, each of which is stored in one byte. The content file specification is shown in Fig. 8. When producing Mobile Web-Com content, the content generation script, the files used by the content and the header generation program are placed in the same directory before running the script.

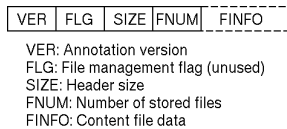


Fig. 8: Content file specifications

4. Implementation of Mobile Web-Com

Mobile Web-Com was implemented in the following environment:

- BREW SDK v3.1.2 Ja
- BREW SDK Tools v1.0.1 (Japanese version)
- Microsoft Visual C++ 6.0
- ARM Developer Suite

This section discusses the implementation of Mobile Web-Com.

4.1 Content File Processing

The content consists of headers and content data compressed and joined together, and the content is expanded into the individual content files according to the header files. The procedure for the expansion of content files is shown below.

1. Read in version from the first byte of header
2. Read in size of file header from the second byte of header
3. Read in number of stored files from the third byte of header
4. Repeat the following steps for each stored file:
 - Obtain the length of the filename by reading one byte.
 - Obtain the filename by reading the file.
 - Obtain the size of the file corresponding to this filename by reading two bytes.

With this procedure, the filenames and file sizes are confirmed, and the compressed files are expanded inside the mobile phone.

4.2 Reading in Annotations

Annotations are allocated on one instruction per record basis and managed as linear lists, and are read into the

Mobile Web-Com system from the annotation files expanded inside the mobile phone according to the following procedure:

1. Read in a 1-byte instruction.
2. If the instruction is a drawing instruction, read in a further 16 bytes of data.
3. If the instruction is a scroll instruction, read in a further 7 bytes of data.
4. Add this data to the record and return to step 1.

4.3 Expansion of background images

To enable smooth scrolling on mobile phones, the background images are connected together and a region is saved as a single image. An advantage of this approach is that the image can be scrolled quickly. However, since this involves saving a bitmap region equal in size to the entire image, it consumes $480 \times 960 \times 4 = 1843200$ bytes of memory.

4.4 Drawing annotations

As in Section 4.3, a region of the same size as the background image is saved for the drawing of annotations, and only annotations are drawn in this region. This region is prepared with a transparent color, and when a color setting flag is set in an annotation record, the line segments are drawn in the specified color. Drawing is otherwise performed in black. Fig. 9 summarizes the drawing of annotations.

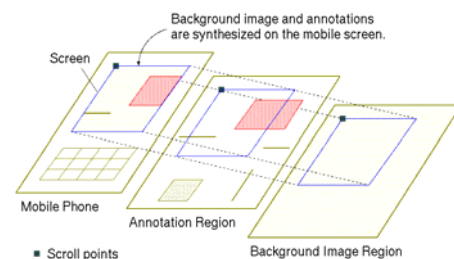


Fig. 9: Drawing of annotations

4.5 Playing back content

The content playback functions in Mobile Web-Com are as follows.

- (1) Playback: The content is played back by displaying the annotation drawing region superimposed on the background image region. Timer processing is performed every `DRAW_TIME` seconds, and processing is performed on annotation records that have not been processed prior to the point reached by the timer. This means that the best drawing efficiency is obtained when the time threshold value discussed in section 2.2 is equal to `DRAW_TIME`.

- (2) Fast forward: Pressing the right arrow key on the mobile phone causes the annotation and voice to move forward by FORWARD_TIME seconds. After performing processing to advance the annotation playback time, the corresponding position in the voice data is retrieved and synchronous playback is resumed, thereby implementing a fast-forward function.
- (3) Rewind: Pressing the left arrow key on the mobile phone causes the annotation and voice to move back by REWIND_TIME seconds. This is implemented by drawing the annotations in a transparent color in the annotation drawing region, and by retrieving the corresponding position in the voice data.
- (4) Pause: Pressing the mobile phone's center key pauses the playback. Pressing the center key again causes the playback to be resumed.
- (5) Annotation jump function: Since mobile phones are only able to display content in a narrow region, the user may not notice the annotations in cases where the annotation is drawn at a position that is separate from the position currently being displayed. To avoid this problem, pressing a key on the mobile phone causes the annotation currently being written to a position at the center of the display.

5. Experimental Results

To confirm the operation of this system, we performed tests using a simulator and an actual mobile phone. The execution environment of the server used in these tests is summarized in Table 7, and Table 8 shows the content data used in the tests. The content in Table 8 is all related to junior high school math classes.

Table 7: Server execution environment

CPU	Intel Celeron 700MHz
Memory size	512Mbyte
Software	Apache2.0.54

Table 8: Content data

		Content 1	Content 2
Play back time (min)		3.3	7.75
Size (Kbyte)	Content file	19	21
	HTML image file	23	44
	Voice file	262	445
	Total file size	304	510
	Compressed file size	179	260

5.1 Simulator Tests

The execution environment of the client simulator used in these tests is summarized in Table 9. In particular, it implemented the following functions:

- Downloading of content
- Play back/pause content
- Drawing of annotations in fast forward and rewind
- Annotation scroll function

Due to the simulator specifications, it was not possible to use functions for fast-forwarding or rewinding the voice data.

Table 9: Mobile Web-Com simulator execution environment

CPU	AMD Athlon 64 Processor 3000+
Memory size	512Mbyte
Simulator	BREW SDK v3.1.2 ja
Machine type	W43H
Network speed	cdma2000 1xEV-DO (up/down:144kbps/2.4Mbps)

Table 10: Time required for downloading content in the simulator

Content	Time (sec)
1	21
2	36

Table 10 shows the results of content download tests on the BREW simulator. Ten test subjects were asked to complete a questionnaire on the use of Mobile Web-Com. The items on this questionnaire are listed in Table 11. Based on this questionnaire, the test subjects were highly appreciative of the operating response of the system and the way in which annotations were drawn when fast forwarding and rewinding the content. On the other hand, they were less impressed by the timing of the voice content and annotations or by the content download times. Comments received from the test subjects included the following points:

- The annotations look more like points than line segments.
- The HTML image in the background is difficult to make out.
- Different blocks of content have different voice audio volume.

Table 11: Questionnaire items

No.	Assessed item
1	Is the HTML content easy to make out?
2	Are the annotations easy to make out?
3	Are the annotations and voice content properly synchronized?
4	Are the content download times acceptable?
5	Are operations sufficiently responsive?
6	Can the voice content be heard properly?
7	Are annotations drawn satisfactorily during fast forwarding and rewinding?

5.2 Tests on Actual Equipment

We made improvements to the download parts of the Mobile Web-Com program used in the simulator, and we then tested the system on actual equipment. In these tests, we measured the amount of time between the selection of content from a content list and the point where the downloading of this content is completed, and we checked the content playback status.

As a result, we found that the voice content playback, pausing, fast forwarding, rewinding and scroll playback functions all operated without problems. However, there were some cases where content failed to download. For content 2, Table 12 shows the amount of time between the selection of content from the content list screen and the point where the screen indicates that the downloading is completed. The download time was approximately 10 seconds for 250 kilobytes of content, thereby greatly alleviating the download time issues experienced with the simulator program. Fig. 10 shows the content playing back on real equipment.

Table 12: Download times of content 2 on real equipment

Machine type	Time (sec)
W43H	7~10
W43CA	8~11



Fig. 10: Content playing back on a mobile phone

6. Conclusion

In this paper, with the aim of developing and checking the operation of an e-learning system for mobile phones, we have migrated the playback functions of a PC-based e-learning system called Web-Com so that they can function on mobile phones running the BREW environment. Mobile Web-Com is the first system created on mobile phones according to the original concept. As a result, we were able to implement various annotation and voice functions by assimilating the differences between the PC and mobile phone environments. In particular, in tests on actual mobile phones, we were able to implement screen scrolling during the playback of content and the drawing of annotations during fast forwarding and rewinding without any problems. However, there is still insufficient stability in the downloading of content on real equipment. For the stable provision of content, it is hoped that further improvements can be made to the content acquisition functions.

The current Mobile Web-Com system does not have any LMS (learning management system) functions such as functions for the administration of learning permissions, educational materials or users. LMS functions also include functions for the management of student grades, publishing of teaching materials on the Web, and bulletin board/chat room functions. In Mobile Web-Com, it is hoped that functions such as authenticated connection to the content management server, user management and student learning progress management will also be included.

In real classrooms, communication and mutual relationships between the teacher and each individual student are easily formed, making real classrooms effective for the transfer of knowledge and the formation of learner communities. In virtual classrooms, students can sometimes feel isolated and remote from the rest of the class, but such problems can be avoided by enabling the students to engage in cooperative activities. In the future, we will need to subject Mobile Web-Com to further assessment from the viewpoint of educational effectiveness.

Acknowledgments

We would like to thank KDDI Corporation for their support with regard to the use of mobile phones in this study. This research was partly funded by a Grant-in-Aid for JSPF Scientific Research (B18300027).

References

- [1] Ministry of Economy, Trade and Industry, Business-Affairs Information Policy Office, Information Processing Promotion Division supervised, Japanese e-Learning Consortium ed., E-Learning White Paper, 2006/2007 Edition, Tokyo Denki University Press, July 2006.
- [2] L. Zhang, Y.-T. Zhuang, Z.-M. Yuan, and G.-H. Zha, "A web-based examination and evaluation system for computer education," Proc. of the Sixth IEEE Int'l Conf. on Advanced Learning Technologies ICALT'06, pp.120-124, July 2006.
- [3] J. Kahan and M.R. Koivunen, "Annotea: an open RDF infrastructure for shared web annotations," Proc. WWW10 Int'l Conf., pp.623-632, 2001.
- [4] S. Ramachandran, and R. Kashi, "An architecture for ink annotations on web documents," Proc. Seventh Int'l Conf. on Document Analysis and Recognition ICDAR 2003, pp.256-260, 2003.
- [5] M.A. Chatti, T. Sodhi, M. Specht, R. Klamma, and R. Klemke, "u-Annotate: an application for user-driven freeform digital ink annotation of e-learning content," Proc. of the Sixth IEEE Int'l Conf. on Advanced Learning Technologies ICALT'06, pp.1039-1043, July 2006.
- [6] K. Hiraki, T. Yonekura and S. Shibusawa, "Web-Com": interactive browser for web-based education," Trans. IEICE, ED Special Issue on Cyber Worlds, vol.E88-D, no.5, pp.926-937, May 2005.
- [7] K. Hiraki, T. Yonekura and S. Shibusawa, "An evaluation of Web-Com: web-based education system," Proc. of Int'l Conf. on CyberWorlds CW2005, pp.229-236, Nov. 2005.
- [8] K. Papanikolaou and S. Mavromoustakos, "Critical success factors for the development of mobile learning applications," Proc. of the 24th IASTED Int'l Conf. on Internet and Multimedia Systems and Applications, pp.19-24, Feb. 2006.
- [9] R.S. Cobcroft, S. Towers, J. Smith, and A. Bruns, "Mobile learning in review: opportunities and challenges for learners, teachers, and institutions," Proc. of the Online Learning and Teaching Conf., pp.21-30, Nov. 2006.
- [10] D. Parsons, H. Ryu, and M. Cranshaw, "A study of design requirements for mobile learning environments," Proc. of the Sixth IEEE Int'l Conf. on Advanced Learning Technologies ICALT'06, pp.96-100, July 2006.
- [11] H. Qi, M. Wang, R. Tong, R. Shen, J. Wang, and Y. Gao, "The design and implementation of an interactive mobile learning system," Proc. of the Sixth IEEE Int'l Conf. on Advanced Learning Technologies ICALT'06, pp.947-951, July 2006.
- [12] D. Averberg, J. Boppert, K. Holzweissig, T. Loke, T. Riemann, and J. Magenheimer, "Mobile delivery server(MDS) --- a solution for resolving problems and limitations in mobile e-learning scenarios," Proc. of ICNICONSMCL, 2006.
- [13] K. Nakabayashi, T. Hoshide, M. Hosokawa, T. Kawakami, and K. Sato, "Design and implementation of a mobile learning environment as an extension of SCORM 2004 specifications," Proc. of the Seventh IEEE Int'l Conf. on Advanced Learning Technologies ICALT'07, pp.369-373, July 2007.
- [14] T. Laakko and T. Hiltunen, "Adapting web content to mobile user agents," IEEE Internet Computing, vol.9, no.2, pp.46-53, March/April 2005.
- [15] M. Kamata, S. Shibusawa, T. Yonekura, "e-Learning system "Mobile Web-Com" for mobile phones," Proc. of Int'l Conf. on Computational Intelligence and Security CIS2006, pp.886-890, Nov. 2006.
- [16] QUALCOMM BREW, <http://brew.qualcomm.com/brew/en/>
- [17] BREW Lab., <http://www.brewjapan.com/lab/index.html>
- [18] Vorbis.Com, <http://www.vorbis.com/>
- [19] PureVoice, <http://www.cdmatech.com/solutions/products/purevoice.jsp>



Susumu Shibusawa received the B.S. and M.S. degrees in Physics from Niigata University in 1974 and 1976, respectively, and Ph.D. in Information Engineering from Kyoto University in 1991. He joined Department of Computer and Information Sciences, Ibaraki University in 1992, and he is now an Associate Professor at Ibaraki University.



Tatsuhiro Yonekura received the B.S. and M.S. degrees from Nagoya University in 1979 and 1981, respectively. He then joined Yamatake Honeywell, working on development of industrial computer system till 1990. He received the Ph.D. degree in Information Science from Nagoya University in 1991. From 1993 he was an Associate Professor, and from 2005 he is a Professor in Department of Computer and Information Sciences at Ibaraki University.

Michitoshi Niibori received the B.S. degree in Computer and Information Sciences from Ibaraki University in 2005. He is a cofounder of Learning-i Co. Ltd., and now a CEO at Learning-i.

Munchisa Kamata received the B.S. and M.S. degrees in Computer and Information Sciences from Ibaraki University in 2005 and 2007, respectively. He is now with Hitachi Business Solution Co. Ltd.