PAPER

Advanced Content Authoring and Viewing Tools Using Aggregated Video and Slide Synchronization by Key Marking for Web-Based e-Learning System in Higher Education

Sila CHUNWIJITRA^{†a)}, Student Member, Arjule JOHN BERENA^{††b)}, Hitoshi OKADA^{††c)},
and Haruki UENO^{††d)}, Nonmembers

SUMMARY In this paper, we propose a new online authoring tool for e-Learning system to meet the social demands for internationalized higher education. The tool includes two functions – an authoring function for creating video-based content by the instructor, and a viewing function for self-learning by students. In the authoring function, an instructor creates key markings onto the raw video stream to produce virtual video clips related to each slide. With key markings, some parts of the raw video stream can be easily skipped. The virtual video clips form an aggregated video stream that is used to synchronize with the slide presentation to create learning content. The synchronized content can be previewed immediately at the client computer prior to saving at the server. The aggregated video becomes the baseline for the viewing function. Based on aggregated video stream methodology, content editing requires only the changing of key markings without editing the raw video file. Furthermore, video and pointer synchronization is also proposed for enhancing the students' learning efficiency. In viewing function, video quality control and an adaptive video buffering method are implemented to support usage in various network environments. The total system is optimized to support cross-platform and cloud computing to break the limitation of various usages. The proposed method can provide simple authoring processes with clear user interface design for instructors, and help students utilize learning contents effectively and efficiently. In the user acceptance evaluation, most respondents agree with the usefulness, ease-of-use, and user satisfaction of the proposed system. The overall results show that the proposed authoring and viewing tools have higher user acceptance as a tool for e-Learning.

key words: aggregated video, authoring tool, video-based content, key marking, WebELS learning

1. Introduction

Due to the rapid growth of information and communications technology (ICT), e-Learning systems have become widely used in delivering various educational programs. The systems provide several advantages that can be easily achieved at a lower cost and higher quality on a global scale [1]–[3]. While social demand for internationalized educational program is continuously increasing, the role of universities is to move the learning environment from the traditional method to the distance learning using ICT [4]. Higher ed-

ucation students and company employees are continuously acquiring more advanced knowledge. Unlike undergraduate programs where course management and automated student assessment are typical features, higher education students utilize ICT materials to support their learning and research activities. Based on characteristics of higher education, e-Learning system is strongly required because traditional classroom-based education cannot answer this kind of distance learning and social demand [5], [6].

Methodologies of e-Learning service have been changing in recent years. Video-based content is increasingly popular not only because it has more attractive visualization but also effective in stimulating learners' interest [7]. With the advancement of Internet-working technologies, streaming video can offer exciting opportunities for online teaching and learning. This form of technology brings courses alive by allowing online learners to use their visual and auditory senses to learn new concepts. To this end, many methodologies and products have been proposed and are being used to support video-based learning content [8]–[13].

Instructors aim to provide learning content to their students with high-quality user experience and user-friendly operations by designing simple processes. Most authoring tools focus on the online content since e-Learning systems are usually designed for server-client communication. Some authoring tools are designed for creating offline content, while several authoring tools can completely author by themselves. However, some tools need external applications to manage audio or video before creating the learning content, and re-editing the video-based content is difficult or impossible with some products.

Recently, some e-Learning systems have been initiated to support academic institutions in rural areas [14], [15]. For example, the Sahara Solar Breeder (SSB) Project is trying to establish video-based learning contents to be shared between Japan and Algeria [16]. Also the UNESCO Collaboration Project has proposed to create a new e-Learning archive and to share it within the collaborative countries including Japan, China, Thailand and Algeria [17]. In the project, streaming-based content authoring and viewing modules are requested, especially in areas with low-speed Internet. Traditionally, lecture presentation slides and recorded video are kept for archiving purposes. However, these archived raw materials have not been useful for on-

Manuscript received December 3, 2012.

Manuscript revised March 20, 2013.

[†]The author is with The Graduate University of Advanced Studies, Kanagawa-ken, 240–0193 Japan.

^{††}The authors are with National Institute of Informatics, Tokyo, 105–0123 Japan.

a) E-mail: sila@nii.ac.jp

b) E-mail: berena@nii.ac.jp

c) E-mail: okada@nii.ac.jp

d) E-mail: ueno@nii.ac.jp

DOI: 10.1587/transinf.E96.D.1754

demand learning, and they are just keep for years without being re-used. A combined content of synchronized video and presentation slide is strongly requested to create useful learning contents. It is very desired and a good opportunity to re-use archives which can be shared among learners in the project. However, the network infrastructure in these countries is not good enough for applications requiring high-speed network. Hence, it is hard to use the video materials for e-Learning contents. Furthermore, several users' technological requirements are requested, e.g., (1) convenient and quick to create and edit contents, (2) able to save editing stages and able to continue editing in the last saved editing stage, (3) employ existing raw video files to re-use as new learning contents in suitable method, and (4) able to use applications without software installation.

A web-based e-Learning system known as WebELS was proposed to support flexibility for higher education especially in PhD education [18]. WebELS provides an easy authoring tool to manage the learning contents based on Java technology for cross-platform support. This tool is suitable for slide-based content including audio and cursor synchronization. To enhance the capability of WebELS in supporting e-Learning in higher education, we proposed a new authoring tool which supports online video-based contents, and can co-exist with audio-based contents. The new authoring tool is implemented on a cloud computing environment to ensure reliability and scalability of the system. Cloud computing is a strategic technology that uses the Internet and central remote servers to maintain data and use applications without personal software installation. This strategic system allows more efficient computing by centralizing storage, memory, processing and bandwidth [19].

In this paper, we proposed a new online authoring tool for WebELS e-Learning system to manage learning content that consists of presentation slides and recorded raw video. The tool utilizes key marking to produce the aggregated video stream which is used in viewing function instead of real video stream. With key markings, some parts of the raw video stream can be easily skipped, and using the aggregated video stream provides easier editing. We also proposed video and pointer synchronization methodologies to enhance the system efficiency by providing the student a pointer guide. The authoring and viewing functions are also optimized to support low-speed Internet, cross-platform and cloud computing to widen the scope of users and break the usage limitation. An adaptive video buffering method is applied to the viewing function to keep the smooth video playback. Clear user interface and simple processes were considered in the design. This system is utilized to support distance learning and higher education among universities, and to meet social demands for internationalized education.

2. Review of Related Studies and Technologies

There are a lot of available technologies and products, and some have similar and overlapping functions, thereby, we conducted the literature review related to e-Learning and

authoring system for video-based content. The following subsections discuss the technologies and products according to the three aspects, such as learning content characteristics, authoring methodology, and the types of e-learning platform.

2.1 Learning Content Characteristics

The prevalent characteristics of learning content can be categorized in three groups; (1) general web-based, (2) slide-based and (3) video-based. General web-based learning content is used as a standard e-learning system since web technology was first initiated. It uses standard HTML elements such as text, images, video streams and links for making content. Many CMS systems support and use SCORM for content sharing among other systems, such as Moodle, and Joomla LMS.

Slide-based learning content is composed of slides with embedded objects, such as text, image and others. Some tools can create the learning content from the slide presentation file. For instance, iSpring is a tool used to convert PowerPoint presentations into Flash video files [20]. A Lecture of Demand (LoD) toolset is proposed for producing reusable web-based multimedia presentation [21]. It uses PowerPoint presentation file and video file. The LoD system requires external software to encode and consolidate all video files into a single real video file. They use SMIL to save output and playback by Real Player.

Video-based learning content becomes interesting and popular. The main characteristic of video-based content is to integrate video stream and slide presentation into one learning content. The video stream is used as a baseline of a learning content. Presentation slides are automatically changed by video timing. However, there is a weakness in this method, i.e., it is difficult to skip some part of video that contains noises or garbage data. Todai-eTEXT [13] and Mediasite [12] are examples of video-based learning content.

2.2 Authoring Methodology

Several methodologies of authoring tool for synchronizing video and slide are proposed. One method is processed-based, which we may further classify into pre-processed and post-processed. In case of pre-processed, such as Adobe Authorware and Toolbook, editor must prepare the required data before making the learning content. For example, video clips must be already captured or cut from the raw video stream using another tool. The system supports SCORM standard. In the post-processed method, editor can create the content without video clips preparation such as in Todai-eTEXT [13]. The authoring tool can help the editor cut the raw video into video clips directly.

Another interesting method is the real-time learning content creation. Nael Hirzallah proposed an on-the-fly creation of the video content from the classroom lecture [22]. He developed the Slide Generation algorithm (SG) used to detect the change of slide by capturing the image from camera. This system however requires high-end computer per-

formance for saving and comparing algorithm. The output is saved in SMIL format. Also, Yi-Chun Liao proposed a story-based editing and browsing system with the automatic video segmentation [23]. The system combines video, audio and screen output in real-time, and inserts the interaction mark to control the presentation slide. It also uses hypervideo for his authoring. Other authoring products – CamStudio, Mediasite and Capture Station – allow real-time recording of screen including the presentation and audio activity on the computer to create video-based content. Mediasite and Capture Station allow editing the live presentation before publishing, while CamStudio needs to re-record after the content was created. The disadvantage of Capture Station is that it requires special system and hardware in order to work.

2.3 e-Learning Platform

There are three main types of e-Learning platform: (1) the standalone system, (2) the server-client system, and (3) cloud computing system. Several applications are working as a standalone system, such as iSpring, CamStudio, etc. They do not need the network infrastructure but requires installation of software application in user's computer.

Most e-Learning system is server-client based platform. For example, the Web-based Synchronized Multimedia Lecture (WSML) framework, which was initiated for supporting the Web-based Chinese classroom in Taiwan [24]. WSML system provides an authoring tool to record the temporal and spatial relationships among media. Another platform in Singapore is still on the implementation stage. They have started an online community of practice for teachers and designed an online platform where teachers can share vivid images and videos of their teaching practices with other teachers [25].

The implementation e-Learning on a cloud computing platform has its peculiarities and needs a specific approach. Bo Dong presented an e-Learning ecosystem based on cloud computing infrastructure [19]. The benefits of the system are reliable, flexible, and cost-efficient. The system also has mechanisms to guarantee the teaching and learning activities, and the stability of the learning ecosystem. Paul Pocatilu measured the positive impact of using cloud computing architectures upon e-Learning solution development [26]. Result shows that cloud computing system can reduce the cost of infrastructure maintenance and risk of hardware failure of e-Learning system.

3. System Design and Implementation

The system design concepts described in this section covers the system framework, the aggregate video by key marking concept, the pointer movement and video synchronization concept, and data structures. Moreover, we discuss the implementation of the system that includes the system architecture, the authoring function for instructors and the viewing function for students with emphasis on supporting distance learning in higher education and usage in low-speed

network.

3.1 Design Concepts

3.1.1 System Framework

Figure 1 shows the workflow of the proposed system which is designed for supporting client-server operations. The client side is a web-based application which can be accessed by any web browser by both instructor and students. We defined the common containers for communicating and sharing data between client node and server system. These are the three files used to support the infrastructure. First, a *metadata file* that is used to retain the content definition, such as content title, description, references, etc. The benefit of using metadata file is that it is a simple text file. It is small and fast to transfer between client and server. Second, a *content template* that is utilized to keep the synchronization data created by the editor. Several data are contained in the content template, such as video index, key mark values, slide index, and pointer actions. Finally, a *learning content package* which is the output container of learning content. It includes metadata, presentation slides, video stream, and synchronization data.

The workflow shown in Fig. 1 can be separated into two functions – an authoring function for instructors and a viewing function for students. The authoring function workflow consists of eight main steps for creating a video-based content: (1) requesting the editing tool from the server, (2) preparing the editing tool environment, (3) uploading source files to the server, (4) converting source files to required formats (flash and image), (5) creating *metadata file* after successful file conversion and automatically sending the data to the client, (6) synchronizing slide and video, and creating *content template*, (7) saving and uploading the content template to the server, and (8) automatically generating *learning content* and storing at the server. On the other hand, student who has permission to access to the learning content can request to view the content. The viewing function workflow consists of four steps: (1) requesting the viewing tool from the server, (2) preparing the viewing tool environment, (3) automatically receives the metadata content, and (4) automatically plays the complete video and slide synchronized learning content.

In addition, the framework can be applied in a cloud computing environment. It provides efficient computing by centralizing the resources at the server. In authoring and viewing functions, client's computer can keep the data and use applications without software installation.

3.1.2 Aggregated Video by Key Marking Concept

The concept of the proposed authoring tool for managing the video-based content is mainly to synchronize video stream and presentation slides. Our idea is to replace the raw video stream with an aggregated video streams in the viewing function, wherein some unnecessary parts of the raw video stream can be skipped. This technique is a fulfillment and

Fig. 1 Authoring and viewing functions workflow.

Fig. 2 Concept of aggregated video stream and slide synchronization.

answer to the limitations of other methodologies for authoring video-based content as described in Sects. 1 and 2. The technical concept behind content authoring based on aggregated video is shown in Fig. 2. The proposed system utilizes the aggregated video stream with key markings as baseline for synchronizing to the slide series, in contrast to the traditional method of using a raw video stream as the baseline. Key marking is a technique used to produce virtual video clips from the whole raw video stream. There are three important key marks of the structure that defines a virtual video clip, such as (1) *video index* that is used as pointer to refer the raw video file, (2) *key start* that is used as a timing value of raw video file which defines the beginning position of virtual video clip, and (3) *key stop* that is used as a timing value of raw video file which defines the ending position of virtual video clip. After defining all needed virtual video clips by key marking, an aggregated video stream is made up from virtual video clips subsequently.

To obtain a synchronizing data for the video and presentation slides, the aggregated video stream is devised for providing the synchronization process instead of the actual video stream. The aggregated video stream composed of a series of indexed virtual video clips have key marks for start and stop times. The key start and key stop times are utilized to trigger the slide change for synchronizing a slide presen-

tation with video. The aggregated video stream becomes the baseline of the learning content, instead of the actual raw video stream as used in the traditional method.

To accomplish the design concept, the aggregated video stream data is stored in the XML document. Each row contains a series of virtual video clips as follows:

For encapsulated virtual video clip, the structure is:

$$vc = [rv] : [kstart] : [kstop] \quad (1)$$

where

vc - is the virtual video clip

rv - is the video index referred on the raw video stream

kstart - is the key mark start time position on the raw video stream for this virtual clip

kstop - is the key mark stop time position on the raw video stream for this virtual clip

And for encapsulated aggregated video package, the structure is:

$$[vc_1][vc_2] \dots [vc_N] \quad (2)$$

where

vc - is the virtual video clip encapsulated from (1)

Fig. 3 Aggregated video and pointer synchronization structure.

Fig. 4 Data structure of learning content package.

3.1.3 Pointer Movement Concept

It is well-known that video-based content is highly evolving and even more compelling for an e-Learning content [8]. In this paper, pointer movement concept is also proposed to synchronize pointer movement on the slide as the video plays. Pointer object can be positioned in the slide to simulate the presenter's focus of discussion, i.e., what the presenter is talking about. This proposed method is intended to increase the student's desire to learn. It also increases their abilities of proper understanding and retention.

To implement the concept, pointer movement positions are recorded as a series of mouse coordinates (X, Y) that correspond to the aggregated video timing. When the mouse is stationary, it is not necessary to record all the mouse coordinates with the video stream timing. The tool only captures the data when mouse button is pressed and dragged by the editor, thereby minimizing the size of pointer action information. Each slide has its own timeline depending on synchronized video stream. The recorded video and pointer synchronization data are separately stored into an archive. Figure 3 shows methodology of online aggregated video and pointer movement synchronization. Values and structure which need to be recorded are shown as follows:

$$pt = [timing][position] \quad (3)$$

where

pt - is the pointer index

$timing$ - is the aggregated video clip timing in millisecond (ms)

$position$ - is the mouse coordinate (X, Y) at time, timing

3.1.4 Data Structure

Figure 4 shows the data structure of the proposed video-based learning content package. The content package consists of two segments – the header information and the slide chunk series of the presentation. The header segment with six fields contains the content general information, such as title, category, subcategory, abstract, author, reference, raw video file reference and description of content. In the slides chunk segment, each slide is an independent unit with five fields, which contains important slide elements, such as header, image, aggregated video stream, pointer data and slide description.

From the data structure, presentation slide is used as the baseline of content stream. Header information is key data used to reference other data chunks, making it flexible for rearranging the slide sequence that includes the whole data in a chunk. Image data is used for linking to an image file and pointer data also linked to a file that stores pointer action data. Aggregated data are referred to the aggregated video stream that the virtual video clips are encapsulated. Slides data are stored into each file as separate data packages instead of one complete file. It is considered to be the best method for changing the metadata file during editing and

viewing operation.

3.2 System Functions

3.2.1 System Architecture

We highly consider the reliability and flexibility of the system, thus a client-server system is designed. Contents should be stored on the server side or cloud system to avoid the risk from unstable client's computer. Cross-platform usage should be also supported. The functions can be divided into two parts – the client part and the server part.

Client part is the main user interface used by users. Users should keep data and use applications without software installation at their computer. The system should be able to completely support both instructor and student's tasks. The tool helps instructors create learning activities for students anytime and anywhere, and manage the contents without the risk from unstable client's computer. Therefore, we proposed a web-based application which can be accessed via the general web browsers. Adobe Flash technology is the best choice for implementing the user interface, because Flash technology is popular at the moment. It has strong benefits, such as lightweight application and supports cross-platform. It can be plugged-in to any web browsers. Furthermore, Adobe Flash has its own technology for using the video stream format that can be adapted for real time video playing. When viewing the learning content, network connection is always needed for online Flash streaming from the server. Hence, video quality selection function is designed to provide suitable quality content in various network environments without any barriers.

On the other hand, the function on server side is the back-end processes of the proposed authoring tool. The system is performed by Tomcat servlet running on the Linux OS. We decided to use Red5 for managing the video stream data since it is an Open Source Media Server compatible with the Flash streaming content. It supports the RTMP for communicating to Flash clients, which allows clients to dynamically interact with the server to stream audio/video. We also use flash video format (FLV) to contain the video content because it can compress movie file much more efficient than other traditional video formats, and it takes much less buffering time for playing over the Internet. Several functions are designed for operating the editing commands. Source files are uploaded by editor and converted to the required formats. Within the conversion process, there are two steps. First, each slide is converted into one image file (PNG). Second, the raw video stream is converted to flash format (FLV) in three output qualities such as low, medium and regular qualities to support usage in various network environment. In the second step, the system requires high resources and process time depending on the duration of video input stream. After converting the files, metadata file is generated and automatically sent to the editor computer. Finally, the learning content package is generated and stored in the archives at the server when the save command is called.

3.2.2 Authoring Function

Authoring function is used by instructor for managing learning contents. To create a new learning content, source content files, such as video and presentation file must be uploaded to the server. These files are converted to the required formats at the server. After the files have been converted, the tool automatically receives the metadata content from server for preparing the work environment. Subsequently, the editor can synchronize the whole video stream with each slide by means of easy-to-use control buttons. Editor can also manage content, e.g., inserting slide title and description, and adding mouse pointer action to each slide. These operations are done at the client machine, and then uploaded to the server after editing is finished to reduce network traffic.

Figure 5 shows the design of user interface for authoring function. It is divided into six panels, namely (1) Slide Navigator Panel for showing all slide pages and for quick slide changing, (2) Raw Video Panel for displaying raw video stream, (3) Slide Preview Panel for previewing a selected slide, (4) Aggregated Video Preview Panel for displaying an aggregated video which is synchronized to a selected slide, (5) Content Information Panel for showing all details of synchronization slides, and (6) Editing Control Panel for managing learning content. Editing Control Panel contains synchronization tools, remove synchronized video, add blank slide, remove slide, pointer movement management and slide information editor. The authoring tool can help instructor to quickly create an aggregated video package using key marking from the raw video material by simple processes.

Editor can generate pointer movement of each slide which is synchronized with aggregated video stream. The popup interface is separated from the main authoring interface as shown in Fig. 6 because of the limitation of screen. There are three panels, namely (1) Slide Panel for recording the pointer movement, (2) Aggregated Video Panel for displaying an aggregated video of a current slide, and (3) Control Panel for supporting editor to synchronize pointer move-

Fig. 5 Design of user interface for authoring function.

Fig. 6 User interface showing pointer movement synchronization.

ment with aggregated video stream. It is easy to add, remove and preview the pointer movement. To record the pointer movement, editor can press mouse button and move mouse position in the slide panel area while aggregated video is playing. A special mark is used to visualize as pointer movement. The tool records movement positions which are synchronized to the aggregated video timing, and saved the data to the file subsequently.

3.2.3 Viewing Function

Viewing function is used for displaying the learning content. When a student selects a specific content, the tool then automatically downloads the content template in XML file that contains key marking data, such as video index, key start and key stop, that serves as control information to manage the synchronization of aggregated video and slides. Using these key marking data, the RTMP streaming transfers the aggregated video from the Flash streaming server to the client video player, playback starts when the buffered video data has reached a minimum specified amount, and playback continuous until end of the aggregated video. When changing a slide, video streaming is controlled by a corresponding key marking data. With RTMP streaming, it is easier to skip part of the video in creating an aggregated video for video and slide synchronization, and downloading of the entire raw video onto the viewer's video player is not necessary.

The system has three video content qualities for the viewer to support usage in various network environments. During playback, quality controller is utilized to provide suitable video content quality in all network environments. The tool provides the option for students to use video quality optimized for low-speed network environment and to view contents without any inconvenience due to frequent stoppage for buffering. Generally, video content does not continuously play in the low-speed network, and that it looks like frame-to-frame motion in case of direct streaming delivery for short buffering time setting. To enhance the video playback process synchronized for each slide, a progressive downloading technique is considered [27]. With this technique, video stream data is downloaded and buffered as it is received, and then playback starts after a specified mini-

um portion of the data has been downloaded. Progressive downloading requires buffer to hold data for the video player to use at a later time. Therefore, we implemented an adaptive video buffering method to the video player section [28], [29]. The method is utilized to keep long buffering time for each aggregated video clip for smooth video playback in the narrow Internet environment. The video buffering time of the player is automatically adjusted based on the Internet bandwidth and the video bitrate. The adaptive video buffering time equation is shown below:

$$bt' = vl(1 - (bw/vb)) \quad (4)$$

$$bt = \begin{cases} k_{max} & \text{for } bt' \geq k_{max} \\ bt' & \text{for } k_{min} < bt' < k_{max} \\ k_{min} & \text{for } bt' \leq k_{min} \end{cases} \quad (5)$$

where

bt' - is the estimated video buffering time (second)

bt - is the adaptive video buffering time (second)

vl - is the aggregated video clip length (second)

vb - is the video bitrate (kbps)

bw - is the network bandwidth (kbps)

k_{max} - is the maximum buffer time constant (second)

k_{min} - is the minimum buffer time constant (second)

In Eq. (5), we set k_{max} and k_{min} to 60 seconds and 0.1 second, respectively, while bt' is dependent upon the network bandwidth and video clip parameters calculated using Eq. (4). The first condition in Eq. (5) happens when bt' is equal to k_{max} or more as a result of a higher video bitrate than the network bandwidth. At this condition, we keep the video buffering time equal to k_{max} . In the second condition, the video buffering time will be set equal to bt' for its value ranging from k_{max} and k_{min} . This condition also occurs when the video bitrate is higher than the network bandwidth and for all aggregated video clips less than k_{max} . The third condition happens when the value of bt' is less than k_{min} regardless of the length of the aggregated video clip. This means the video bitrate is less than the network bandwidth, such as in the high-speed network environment.

Figure 7 shows a mechanism for video playing with progressive downloading and buffering of real-time data streams. When the viewer starts to view content, the video stream is downloaded and buffered until the video buffer becomes full or has reached bt , and video playback starts. During playback, incoming video data stream is continuously downloaded and buffered until the end of the video stream, and then video playback stops.

Figure 8 shows the viewing function interface design. Students can only view the learning content, but editing is not allowed. The interface consists of four panels, namely (1) Aggregated Video Panel for displaying the aggregated video of a current slide, (2) Slide Panel for displaying the current slide, (3) Content Information Panel for showing slide information, and (4) Control Panel for controlling on-line learning content. Control panel contains quality control, zoom control, toggle view control and full screen mode. Student can toggle view between video and slide panels.

Fig. 7 Mechanism for video playback with progressive downloading technology.

Fig. 8 Design of user interface for viewing function.

Fig. 9 An example of zoom-in feature in the viewing function.

They can also zoom both video or slide contents to examine more closely or in greater detail. Pointer movement is visualized in this panel in case the editor synchronized it with the aggregated video. The pointer mark automatically

moves while the video is playing. An example of zoom-in function is shown in Fig. 9.

4. Evaluation and Results

Based on the system framework, there are two issues that need to be evaluated – the system performance on the server side and the client user acceptance of the total system. On the system performance evaluation, we made a comparison for the conversion times of the source files and video buffering during playback for each quality condition as this would influence the user acceptance. Moreover, we conducted a survey to determine the user acceptance of the system by sending out survey questionnaires to users of the system from four different countries.

4.1 System Performance Evaluation

We prepared a server machine and installed the proposed system onto that server to evaluate the system performance. The server is driven by Linux operating system on the Virtual Machine server with Intel Xeon Dual-Core @2.33 GHz CPU and 4 GB of RAM. We used nine sample contents wherein each content includes source files such as slide presentation and raw video files. The slide presentations are in PDF file format and each has 30 slides. The video streams are in MP4, AVI and WMV formats and have equally the same specifications as in the following: video bitrate is 930 kbps, video frame rate is 25 fps, audio bitrate is 192 kbps, audio sampling rate is 48 kHz, resolution is 640×480 pixels, duration is 600 seconds and average file size is 70 MB.

In the evaluation procedure, the source files per content were uploaded to the server and the duration of conversion process were determined. The duration of uploading is dependent on the file size and the instantaneous network speed during the entire upload process. The source files were automatically converted to the required formats after finishing the upload process. There are two steps in the conversion process - (1) slide conversion from the PDF file to slide images (PNG) which takes shorter time, and (2) video conversion from raw video file to a Flash video (FLV). The video conversion process requires longer time because it needs to decode raw video and encode to FLV output. The system provides up to three video qualities, therefore the duration for the overall process depends on the number of output qualities selected by the author. The audio-video conversion input parameters for the three qualities and the conversion results are shown in Table 1. Low quality results to smaller file size and the fastest conversion time compared to other qualities. For example, if the length of the raw video is 600 seconds, the converted file size for low quality is around 8 MB, while the converted file size is around 30 MB for regular quality. Conversion process for low quality can be done within 60 seconds, whereas for regular quality requires more than 210 seconds.

Furthermore, we evaluated the adaptive video buffering method in various Internet environments for three video

Table 1 Comparison of conversion process for video qualities.

		Qualities		
		Low	Medium	Regular
Audio-Video parameters	Video resolution (pixels)	160x120	320x240	640x480
	Video bitrate (kbps)	100	200	400
	Video frame rate (fps)	25	25	25
	Audio sampling rate (kHz)	11	22	22
	Audio bitrate (kbps)	32	32	64
	Avg. output file size (MB)*	8	14	30
Conversion results	Avg. conversion speed (second)*	60	120	210

* Based on 600 seconds video length

Fig. 10 Adaptive video buffering time of the viewing function.

qualities. First, we measured the streaming bandwidth for the three qualities and made it as the baseline parameters. The average streaming bandwidth for low, medium and high qualities are 150, 250 and 450 kbps, respectively, where each corresponding to the sum of the video bitrate and the audio bitrate. These values represent the minimum bandwidth required for the video playback without necessary buffering before playback starts. Secondly, we used the bandwidth shaper application to simulate various Internet speed and measured the actual video buffering times. Fig. 10 shows the result of implementing the adaptive video buffering method for each video quality on various network bandwidth. We only considered the video buffering time in the measurement, and these values nearly corresponds to the theoretical values when calculated using Eq. (5). Downloading time of video streaming data until the video buffer becomes full was not included since it depends on the network bandwidth during that time.

However, the actual waiting time of the viewer from the start of viewing until the video plays is always longer than the effect of adaptive video buffering time in the real situation. This is because the video player requires to download the aggregated video stream, store it into the video buffer until the buffer becomes full, then video player starts playing the video stream data from the buffer as shown in

Table 2 Average waiting time of aggregated video in various network bandwidth.

Internet bandwidth (kbps)	Avg. video buffering time (second)		
	Low quality	Medium quality	Regular quality
50	70	120	210
100	45	90	180
150	3	50	120
200	1	35	90
250	1	3	70
300	1	1	55
350	1	1	30
400	1	1	15
450	1	1	5
500	1	1	1

* Based on 90 seconds of video length

Fig. 7. Downloading time is dependent on the network bandwidth and video quality being selected. We measured the average waiting time for the viewer which includes downloading and buffering time of aggregated video stream until video playback starts as shown in Table 2. From the result, longer buffering time occurs when the Internet bandwidth is less than 150 kbps for low quality, less than 250 kbps for medium quality, and less than 450 kbps for regular quality. The viewing tool worked well in three qualities by providing smooth and continuous playback, but have longer waiting time in acutely low-speed Internet environment. In case of viewing the high quality video in low-speed network, incoming video data is continuously buffered but eventually buffer data becomes empty during playback. At this instant, video stops and buffering occurs again. As mentioned earlier, these values represents the baseline bandwidth for the three qualities.

4.2 User Acceptance Evaluation

For the user acceptance evaluation on the proposed tool, we prepared the following materials: test server, raw video files and presentation files, sample learning contents, user manual and an online questionnaire. A group of respondents, mainly higher education instructors and graduate students, were selected from the collaboration members of SSB and UNESCO projects in several countries including Japan, China, Thailand and Algeria. In the process, instructors can download the prepared raw video files and presentation files, or can use their own materials for testing the authoring and viewing functions. On the other hand, students can examine the viewing function by using the learning contents authored by the instructors, or can use the prepared sample learning contents.

We used the online questionnaire in conducting the user acceptance evaluation. The questionnaire has 18 questions divided into three main sections - 8 questions on the authoring function, 6 questions on the viewing function, and 4 questions on the overall system. Each section provides questions that measures the user acceptance of the proposed tool based on the three factors, namely, (1) Usefulness, (2) Ease-of-use, and (3) User satisfaction. We use

Table 3 Tally of results.

No.	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
Authoring Function					
1.**	0 (0.00%)	2 (2.74%)	18 (24.66%)	32 (43.84%)	22 (30.14%)
2.*	0 (0.00%)	1 (1.37%)	15 (20.55%)	32 (43.84%)	25 (34.25%)
3.*	0 (0.00%)	1 (1.37%)	17 (23.29%)	33 (45.21%)	22 (30.14%)
4.***	0 (0.00%)	1 (1.37%)	19 (26.03%)	29 (39.73%)	23 (31.51%)
5.**	0 (0.00%)	2 (2.74%)	17 (23.29%)	33 (45.21%)	21 (28.77%)
6.**	0 (0.00%)	2 (2.74%)	22 (30.14%)	27 (36.99%)	22 (30.14%)
7.*	0 (0.00%)	2 (2.74%)	18 (24.66%)	29 (39.73%)	24 (32.88%)
8.**	0 (0.00%)	2 (2.74%)	23 (31.51%)	28 (38.36%)	22 (30.14%)
Viewing Function					
9.**	0 (0.00%)	1 (1.37%)	19 (26.03%)	30 (40.10%)	23 (31.51%)
10.***	0 (0.00%)	1 (1.37%)	23 (26.03%)	30 (40.10%)	19 (31.51%)
11.*	0 (0.00%)	1 (1.37%)	17 (31.51%)	28 (40.10%)	27 (26.03%)
12.***	0 (0.00%)	1 (1.37%)	22 (23.29%)	26 (38.36%)	24 (36.99%)
13.***	0 (0.00%)	1 (1.37%)	14 (30.14%)	35 (35.62%)	23 (32.88%)
14.*	0 (0.00%)	1 (1.37%)	15 (19.18%)	28 (47.95%)	29 (31.51%)
Overall System					
15.**	0 (0.00%)	1 (1.37%)	20 (20.55%)	30 (38.36%)	22 (39.73%)
16.*	0 (0.00%)	2 (2.74%)	23 (31.51%)	28 (41.10%)	20 (30.14%)
17.*	0 (0.00%)	1 (1.37%)	18 (27.40%)	30 (41.10%)	24 (30.14%)
18.***	0 (0.00%)	2 (2.74%)	18 (31.51%)	30 (38.36%)	23 (27.40%)

* : Usefulness, ** : Ease-of-use, *** : User satisfaction

the Likert scale to measure the responses from the respondents [30]. Five ordered response levels are used, such as (1) Strongly disagree, (2) Disagree, (3) Neither agree nor disagree, (4) Agree and (5) Strongly agree, and have corresponding scores as 1, 2, 3, 4 and 5, respectively. Scores were used to determine the user acceptance of the proposed tool based on the three factors mentioned.

We sent out invitation to a group of prospective respondents via email indicating the purpose of the survey, user guide of the system, and the link to the online questionnaire. These prospective respondents consist of IT users who are familiar with computer technologies, and also non-IT users who can use the computer and Internet technologies with little assistance. A total of 73 respondents consisting of 9 instructors and 64 students in the higher education completely answered the questionnaire after using the authoring tool and learning content in actual situations.

Table 4 Summary of evaluation results.

Rating Scale	Functions	User satisfaction (%)	Usefulness (%)	Ease-of-use (%)
Strongly disagree	Authoring	0.00	0.00	0.00
	Viewing	0.00	0.00	0.00
	Overall	0.00	0.00	0.00
Disagree	Authoring	2.74	1.83	1.71
	Viewing	1.37	1.37	1.37
	Overall	2.47	2.05	1.37
Neither agree nor disagree	Authoring	26.03	22.83	27.40
	Viewing	26.94	21.93	27.40
	Overall	27.40	28.05	30.14
Agree	Authoring	39.73	42.92	41.10
	Viewing	41.55	38.36	39.73
	Overall	41.10	39.73	39.73
Strongly agree	Authoring	31.51	32.44	29.79
	Viewing	30.14	38.36	31.51
	Overall	28.77	30.14	28.77

The tally of results for the 18 questions in the questionnaire are shown in Table 3. The results were then analyzed based on statistical mode which represents the value that appears often in the data set. The mode is defined as the element that appears the most frequent number in a given set of elements [31]. There are 8 questions on the authoring function consisting of 3, 4, and 1 question on usefulness, ease-of-use, and user satisfaction, respectively. For the viewing function, there are 6 questions consisting of 2, 1, and 3 questions on usefulness, ease-of-use, and user satisfaction, respectively. For the overall system, there are 4 questions consisting of 2, 1 and 1 question on usefulness, ease-of-use, and user satisfaction, respectively. Responses from each group of questions per factor were combined and the tally of the scores per rating scale were then converted to percentage for easier presentation as shown in the summary of results in Table 4. The result shows that most respondents responded “Agree” to the usefulness, ease-of-use, and user satisfaction of the authoring function, viewing function, and the overall system; except for a tie in viewing function where the same number of respondents responded “Agree” and “Strongly agree” for its usefulness. The values in bold character represents the statistical mode converted to percentage. The descriptive overall results simply show that most of the respondents agree to the usefulness, ease-of-use, and user satisfaction of the proposed tool. The results show that the proposed authoring and viewing tools have higher user acceptance as a proposed tool for e-Learning.

In addition, the questionnaire solicits comments from the respondents optionally. We considered the following helpful comments for enhancing the proposed tool: (1) DOC, DOCX and image file formats should be supported, (2) Viewing function should support an offline mode learning, and (3) The system should integrate more social functions such as forum, chat, etc.

5. Discussions

The proposed authoring and viewing tools have exhibited

numerous advantages as an e-Learning tool for higher education. The total system can help accomplish the purposes of distance learning among universities. It makes the learning activities available anytime and anywhere. In addition, cross-platform and cloud computing are supported to break the barriers in various operating systems and application software installation is not necessary at the users' computer.

The proposed tool can help instructors simplify the process of managing learning contents without external applications required. During content authoring, instructors can preview the synchronized slide and video immediately without saving at the server, thus it is more convenient and saves time for editing the learning content. Some parts of video content which are not related to the slides can be easily skipped by changing the key start and key stop positions of the synchronization data without modifying the source files. Moreover, the proposed tool can help instructors to archive and re-use contents without the risk from unstable computer since the data are stored at the server.

The viewing tool makes the lesson more interesting for students. The tool supports pointer and video synchronization which can make the content attractive and interesting to students. Students can know which part of the slide the presenter is focusing. Students can also learn more in less time, and learning time is never boring. The progressive downloading technique adaptive video buffering time and quality control function were implemented to support usage in low-speed Internet as the progressive download technique. The video buffering time from Eq. (5) can help students to view the content smoothly and continuously until the entire aggregated video clip is finished, or until the video buffering data is emptied. During playback, the incoming video stream is continuously downloaded and buffered. The downloading time is dependent on the selected video quality and network bandwidth during that time. Based on Table 2, buffering occurs when the Internet bandwidth is less than 150 kbps for low quality, less than 250 kbps for medium quality, and less than 450 kbps for regular quality. Low or medium quality content is appropriate choice for students who have low-speed network environment. For smooth video playing for the desired quality, we recommend to provide the minimum required bandwidth or higher for the selected quality.

The performance of the back-end system were evaluated by means of the source files conversion processes. Of the three video output qualities, the conversion of low quality is executed faster, and produced smaller file size than the other qualities. However, the actual conversion speed and output file size may have varied results. They are dependent on the source file input properties such as video encoding, audio encoding, video bitrate, audio bitrate, and resolution.

The proposed tool has been evaluated by 73 respondents who are mainly higher education instructors and graduate students. By using the statistical mode, it was shown that most of the respondents "Agree" to the usefulness, ease-of-use, and user satisfaction of the proposed tool. The overall result shows that the proposed authoring and viewing

tools have higher user acceptance as a tool for e-Learning.

6. Conclusion

In this paper, we proposed a new online authoring tool for e-Learning system to support flexible higher education. This authoring tool is considered to be an "All-In-One" system, since it includes authoring and viewing functions. It can manage learning content consisting of presentation slides and recorded raw video. The tool utilizes the aggregated video stream from raw video with key marking for synchronizing with the presentation slides. It also support video and pointer synchronization to enhance the system functionality. The adaptive video buffer and quality control function were implemented to support usage in low-speed Internet. In evaluating the system, we compared the conversion process time for the three video qualities and verified the adaptive video buffering method by playing the content in various Internet environments for three output qualities. Moreover, we designed an online questionnaire to evaluate the user acceptance of the proposed tool. A total of 73 respondents consisting of 9 instructors and 64 students in the higher education completely answered the questionnaire. Evaluation results show that most of the respondents agree to the usefulness, ease-of-use, and user satisfaction of the proposed tool. The overall result shows that the proposed authoring and viewing tools have higher user acceptance as a tool for e-Learning.

Currently, the project collaboration members are actively using the WebELS system to establish the online learning using the video-based content. In the future, we plan to improve the system to support viewing of content using mobile devices, since it becomes a new trend for e-Learning environment. Moreover, HTML5 technology should be considered to replace the Flash technology.

Acknowledgment

The authors would like to express sincere thanks to all persons who supported the WebELS project of NII Japan, especially to Dr. Vuthichai Ampornarnambeth, Prof. Nobuo Shimamoto and Dr. Pao Sriprasertsuk for contributions in designing and implementing the WebELS system. The project is funded by Science Research Foundation of Japan, Amada Foundation, MEXT, SATREPS of JST and JICA. We express sincere thanks to Genetec Corporation for a collaborative development and business, and the Sahara Solar Breeder (SSB) project for collaborations using WebELS.

References

- [1] J. Boon, E. Rusman, M. Van Der Klink, and C. Tattersall, "Developing a critical view on e-learning trend reports: trend watching or trend setting?," *International J. Training and Development*, vol.9, no.3, pp.205-211, 2005.
- [2] K. Blinco, J. Mason, N. McLean, and S. Wilson, "Trends and issues in e-learning infrastructure development," *A White Paper for alt-ilab 2004 prepared on behalf of DEST (Australia) and JISC-CETIS (UK)*, 2004.

- [3] N. Lawless and J. Allan, "Understanding and reducing stress in collaborative e-learning," *Electronic J. e-Learning*, vol.2, no.1, pp.121–127, Feb. 2004.
- [4] R. Sharpe and G. Benfield, "The student experience of e-learning in higher education: A review of the literature," *Brookes eJ. Learning and Teaching*, vol.1, no.3, pp.1–10, 2005.
- [5] H. Ueno, Z. He, and J. Yue, "Webels: A content-centered e-learning platform for postgraduate education in engineering," *Proc. 13th International Conference on Human-Computer Interaction*, pp.246–255, 2009.
- [6] E.B. Cohen and M. Nycz, "Learning objects and e-learning: An informing science perspective," *Interdisciplinary J. Knowledge and Learning Objects*, vol.2, pp.23–34, 2006.
- [7] T. Hartsell and S.C.Y. Yuen, "Video streaming in online learning," *AACE Journal*, vol.14, no.1, pp.31–43, Jan. 2006.
- [8] H. Borko, "Professional development and teacher learning: Mapping the terrain," *Education researcher*, vol.33, no.8, pp.3–15, Nov. 2004.
- [9] G. Fischer, E. Giaccardi, Y. Ye, A.G. Sutcliffe, and N. Mehandjiev, "Meta-design: A manifesto for end-user development," *Commun. ACM*, vol.47, no.9, pp.33–37, Sept. 2004.
- [10] "Camstudio," <http://camstudio.org/>, accessed Aug., 1 2012.
- [11] "Accordent capture station," <http://www.ivci.com/streaming-accordent-capture-station.html>, accessed Aug., 1 2012.
- [12] "Mediasite lecture capture," <http://www.sonicfoundry.com/webcasting-solutions/lecture-capture>, accessed Aug., 1 2012.
- [13] Todai-eTEXT, <http://todai-etext.ocw.u-tokyo.ac.jp/page.top/index.php>, accessed Aug., 1 2012.
- [14] L. Laschewski, "Innovative e-learning in rural areas: A review," *ICT for inclusive learning: the way forward*, Nov. 2011.
- [15] G. Shinde and S. Jadhav, "E-learning in rural india: Problems, strategies and implementation," *The ICDE International Conference*, Nov. 2005.
- [16] "Sahara solar breeder project (japanese language)," <http://www.iis.u-tokyo.ac.jp/hfujioka/ssb/>, accessed Aug., 1 2012.
- [17] "Webels for unesco collaboration project," <http://136.187.88.193/UNESCO>, accessed Aug., 1 2012.
- [18] V. Ampornaramveth, T. Zhang, A. Hadiana, N. Shimamoto, and H. Ueno, "A web-based e-learning platform for post-graduate education," *Proc. 5th IASTED international conference on Web-based education*, pp.388–393, 2006.
- [19] B. Dong, Q. Zheng, J. Yang, H. Li, and M. Qiao, "An e-learning ecosystem based on cloud computing infrastructure," *Proc. 2009 Ninth IEEE International Conference on Advanced Learning Technologies*, pp.125–127, 2009.
- [20] "ispring," <http://www.ispringsolutions.com/>, accessed Aug., 1 2012.
- [21] F. Bodendorf and M. Schertler, "Producing reusable web-based multimedia presentations," *Interdisciplinary J. Knowledge and Learning Objects*, vol.1, pp.127–142, 2005.
- [22] N. Hirzallah, "An authoring tool for as-in-class e-lectures in e-learning systems," *American J. Applied Sciences*, vol.4, pp.686–692, Sept. 2007.
- [23] Y.C. Liao and C.H. Huang, "Automatic video segmentation and story-based authoring in e-learning," *JSW*, vol.4, no.2, pp.140–146, 2009.
- [24] H.Y. Chen and K.Y. Liu, "Web-based synchronized multimedia lecture system design for teaching/learning chinese as second language," *Comput. Educ.*, vol.50, no.3, pp.693–702, April 2008.
- [25] M.J. Jacobson, H.S. So, H. Lossman, and W.Y. Lim, "Designing an online video based platform for teacher learning in singapore," *Australasian J. Educational Technology*, vol.25, no.3, pp.440–457, 2009.
- [26] P. Pocatilu, F. Alecu, and M. Vetrici, "Measuring the efficiency of cloud computing for e-learning systems," *W. Trans. on Comp.*, vol.9, no.1, pp.42–51, Jan. 2010.
- [27] A. Kőrösi and B. Sézkely, "Peer-to-peer vod: Streaming or progressive downloading?," *Infocommunications Journal*, vol.3, no.1, pp.35–38, 2011.
- [28] M. Narbutt and L. Murphy, "Adaptive playout buffering for audio/video transmission over the internet," *Proc. Seventeenth UK Teletraffic Symposium*, May 2001.
- [29] W. Tu and W. Jia, "Apb: An adaptive playback buffer scheme for wireless streaming media," *IEICE Trans. Commun. (Japanese Edition)*, vol.J88-B, no.10, pp.4030–4039, Oct. 2005.
- [30] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol.22, no.140, pp.5–55, 1932.
- [31] P. Woolf, C. Burge, A. Keating, and M. Yaffe, "Statistics and probability primer for computational biologists," *Massachusetts Institute of Technology*, Sept. 2004.

Sila Chunwijitra was born in 1976. He is a final year student of a Doctor of Philosophy Degree in The Graduate University for Advanced Studies, Tokyo, Japan. He is currently in e-Learning, video conferencing, web applications and open source technologies.

Arjule John Berena was born in 1975. He is a Doctor of Engineering (D.Eng) from Nihon University, Japan. He is currently a post-doctoral researcher at the National Institute of Informatics (NII), Tokyo, Japan. His research fields are e-Learning, video conferencing, web applications and Internetworking technologies.

Hitoshi Okada was born in 1965. He is a Doctor of International Public Policy (Dr. IPP) from Osaka University, Japan. He is currently an associate professor at the National Institute of Informatics (NII), Tokyo, Japan. His research fields are e-Money, e-Commerce and Information Security.

Haruki Ueno was born in 1941. He is a Doctor of Engineering (D.Eng) in Electrical Engineering from Tokyo Denki University, Japan. He is currently a professor emeritus at the National Institute of Informatics (NII), Tokyo, Japan. His research fields are e-Learning Systems, Engineering Education and Knowledge Systems.