# Interactive Media Authoring Systems

Kok Wee Goh, Gangwen Mei, and Shuzhi Sam Ge

Social Robotics Lab, Interactive Digital Media Institute, and Edutainment Robotics Lab Department of Electrical and Computer Engineering National University of Singapore Singapore 117576 Tel.: (+65) 6516 6821; Fax: (+65) 6779 1103 samge@nus.edu.sg

Abstract. Advance in technologies has revolutionized the traditional mode of teaching and learning. Various media authoring systems have been developed in order to enhance the effectiveness of learning and teaching processes by supplementing traditional teaching materials such as text books and lecture notes with interactive digital media. In this paper, we are going to discuss two media authoring systems, KooBits and Web 2.0 Media Publisher, which incorporate creativity, innovation and cutting-edge technologies such as Artificial Intelligence and Asynchronous JavaScript and XML (AJAX). The major part of KooBits system is a client based e-book editor that is specifically designed to enhance interactivity, while Web 2.0 Media Publisher is a server based publishing platform. This paper describes various features, design consideration, implementation of KooBits and Web 2.0 Media Publisher, and compares them with other existing systems in terms of these aspects. Possible future developments for KooBits and some applications for Web 2.0 Media Publisher are also being discussed in the end.

# 1 Introduction

The advent of the information technology has allowed traditional documents to become livelier with the integration of images, animations and other multimedia components into one coherent interactive format [2]. The industry has witnessed one of the earliest manifestations of such documents in the form of powerpoint slides which integrates sound, images and other multimedia tools to assist people in creating better presentation [4,11]. In recent years, industrial trends have moved towards emulating these concepts into electronic books and other document types [13,30].

These new generation document types aim to enhance interactivity between the user and the media. Interactivity is enhanced through three aspects. Firstly, these documents must allow interactive authoring of documents. User must be given the freedom to create and manipulate the document. Secondly, it must also be able to support the authoring of interactive documents. This allows user

Z. Pan et al. (Eds.): Transactions on Edutainment I, LNCS 5080, pp. 206-230, 2008.

to manipulate different interactive media files in one interface. Lastly, it must also allow different authors of the same document to interact [3].

Interactive media authoring systems provide a viable solution to the problems discussed. A multimedia document aims to convey a message to its audience in an interactive and creative way [14]. Unlike traditional static documents, the multimedia support dynamic loading of data based on user input and generate required content in user demand. This allows user to interact with the system. In the past, to create a multimedia document requires the user to possess a certain amount of basic software knowledge. The user needs to learn some software skills to put different media files together. Therefore, it usually takes a user a significant amount of time to develop a multimedia document. Developers soon realized the potential in such documents and devoted efforts into cutting down development time. As a result, various software applications which allow the users to embed other types of media files such as images and videos into the documents were developed. A notable example of such software is the Microsoft PowerPoint. In Microsoft PowerPoint, users are able to import images into the software and allow limited animations. In view of the long development time needed, Hardman, Rossum and Bulterman have also developed a "Structured Multimedia Authoring" to address this problem [9].

This paper presents and discusses two separate media authoring systems – KooBits and Web 2.0 Media Publisher, which rely on two sets of technologies. KooBits contains a client-side application (ebook editor) that requires local installation on the clients' machine and a distributing network for the authors to create and exchange their interactive documents. There is no web 2.0 technology involved in the KooBits system. However, for Web 2.0 Media Publisher, this system makes use of web 2.0 technology to allow the users to author, store and publish their interactive documents completely on the server. In recent years, consumers need more customization that allows a higher level of manipulation on these multimedia documents. KooBits System had been developed as a platform with client side interactive e-book editor and server-side e-book publisher. It has since been introduced to the market to meet this demand [12]. Various novel concepts and features have been introduced with the release of KooBits:

- 1. As an interactive authoring tool, it significantly cut down the time taken to create dynamic multimedia content. This is in sharp contrast to traditional multimedia software which needs a much longer time to create the content. This is achieved by having a deep understanding on the cognitive process of user interaction and smooth integration of various multimedia types.
- 2. Various types of media files such as images and animations are combined in one single coherent interface and can be embedded in the electronic book. Conventionally, different software applications are used to support different types of media files. This makes integration of various media files very difficult. By putting everything in one coherent interface, this will greatly improve the user experience.
- 3. With inbuilt artificial intelligence module, the software is capable of interpreting the meaning of the user's text input and automatically generates

and modifies the graphics content. This is the revolutionary technology introduced by KooBits. With this feature, writing electronic books have transformed from a static, trivial task into one that is vibrant and interactive.

This process greatly enhances user experience for both the author and readers, thereby helping the author to convey ideas more clearly through visualization of the text content. It improves communication between the author and his readers. KooBits help users to clarify abstract concept and visualize the conceptual objects based on his input. Recently, rapid development of infrastructure has resulted in bigger bandwidth. Hence, server side technologies become more mature to support applications. As such, client side applications slowly shifted to server side for better communication. With the emergence of AJAX technology and Web 2.0 concepts, dynamic internet applications can be developed independent of console-based technology. This minimized data transfer and resulted in a more efficient system. Systems developed using this technology do not require the users to install additional software on the client side machine. Based on these concepts, Web 2.0 Media Publisher has been developed as a online media authoring system that allows the users to publish documents created using KooBits or other systems. This paper will first discussed the client based software Koobit which enhance interactivity by providing interactive authoring of documents and authoring of interactive documents. Discussion will then focus on the Web 2.0 Media Publisher which promote interactivity by allowing multiple authors to edit the same document concurrently.

# 2 Design Consideration

Building these media authoring systems requires various rigorous considerations. In order to make the application performing consistently across platforms, subtle details are scrutinized based on research done [21].

# 2.1 Browser Compatibility

In the design of web applications, browser compatibility is always a major issue. The three major internet browsers used at the present moment are Internet Explorer, Fire Fox and Safari. Different browsers interpret the HTML/CSS/ JavaScript code in a slightly different way causing the webpage rendered on the computer screen to appear differently. Currently a popular solution to this problem is to use a conditional statement to check the type of browser the user is using and deliver the appropriate content to the browser accordingly. However, the drawback of this solution is that it significantly increases the size of the javascript program resulting in a longer loading time. For a complicated web application, this method tends to slow down the connection speed and thus reduce the responsiveness of the application.

The situation for mobile browser is even more chaotic. To develop successful content, one needs to consider the diversity of hardware in the market. As different mobile phones use operating system and different browsers, one needs to understand the technical limitations of these mobile browsers. Besides differences between hardware and operating systems between brands of mobile phone, one also needs to note that differences exist even within phones of the same brand. This difference is especially significant when one compares an old model phone with the latest model. As such, to develop the system for the mobile platform, catering to this diversity of hardware is a must. Typically, one needs to identify the model of phone that is requesting for the data before the server can send the appropriate content to phone. When a browser requests for content, it will send a HTTP request to the server. Hence, based on the HTTP request, one can interpret it and determine the model of the mobile phone. Based on the model of the phone, its capability can then be determined by comparing the particular model with a database (a collection of the capabilities of all the phone models in the world). By doing that, appropriate content can be send. The downside, however, is that determining the capabilities of the mobile phone takes time and will slow down the download speed. However, until a better solution is available in the market, this is currently the best possible solution in the market.

### 2.2 Server-Side Technology

Server-side technology is used to allow a Web server to serve dynamic Web content. Dynamic Web content requires the Web server to perform additional processing of the corresponding request to generate customized response. Server-side technology focuses primarily on delivering dynamically generated HTML documents to a user Web browser while the user interacts with a website. Various server side technologies are discussed in deciding the right technology for the system.

- 1. Active Server Pages (ASP) is a server-side technology developed by Microsoft. ASP uses embedded scripts in HTML documents to inform the Web server to generate the necessary HTML data dynamically. ASP supports multiple scripting languages like VBScript, JavaScript and PerlScript, with VBScript as the default scripting language. ASP includes support for Microsoft-specific Web technology like ActiveX, which provides useful functionality like file manipulation. ASP applications can only be developed and deployed on the Windows platform and Web servers like Internet Information Server (IIS) and Personal Web Server (PWS). Third party applications are required to port ASP to other platforms.
- 2. Macromedia ColdFusion, formerly from Allaire, uses a different approach to enable Web servers to generate dynamic content. Instead of embedding some programming language into the HTML document, ColdFusion uses its own proprietary ColdFusion Markup Language(CFML), which is a tagbased language. ColdFusion has about 100 built-in tags to access variables, databases and build conditions. In addition, ColdFusion allows the creation of custom tags, which can be created with CFML, or using C++ and ColdFusion API. ColdFusion documents end with a .cfm extension. The Web server hands over any .cfm document to the ColdFusion server. The ColdFusion server processes the document and sends the output back to the Web server,

which then sends it to the Web browser. While this approach makes it very easy to build Web applications, if a complex Web application is required, C++ is required to link to the ColdFusion server. This creates problems like memory-leak in complex codes that are not well-written. And since the code is running all the time inside the ColdFusion server, this can easily lead to disastrous consequences like crashing of server due to exhaustion of memory.

3. Personal Home Pages (PHP) is a server-side technology developed in the Open Source community. It is similar to technology like ASP and JSP in that PHP consists of a set of additional code tags placed inside the HTML documents. However, PHP is unique in that it is a language developed purely to serve Web pages rather than being based off an existing language such as Visual Basic or Java. PHP provides support for many kinds of Web applications, including database access, graph plotting and networking. PHP is available freely and supports a wide range of platforms.

# 2.3 Client-Side Technology

Client-side technology is used to add dynamic behavior to HTML content by executing certain routines on the Web browser. Client-side technology helps to ease the load on the Web server and shortens the response time for each request for dynamic content. This section looks at some of the common client side technologies available.

- 1. AJAX (Asynchronous JavaScript + XML) is a relatively recent web programming technology developed. With certain parts being known as Dynamic HTML and remote scripting, AJAX magically stretches a bunch of trivial old technologies such as JavaScript and CSS well beyond their original scope, and turns them into the powerful tool for building complicated web service. The public attention is attracted to AJAX when some major web application developer, such as Google, shows the general public what AJAX can do. The classic type of websites and web application is that visitor browse through the web page on the server one by one. In contrast, one of the main concepts in the AJAX technology is to put in some JavaScript control logics inside the web pages to monitor user actions and alter the HTML code and CSS styles. By doing so when user access the internet, what the server returns are not separate web pages, but a client application for the first access, then simply data in the following communication.
- 2. Macromedia's Flash is a system which is able to create interactive movies in a compressed vector graphics format, which is one of the strength of flash because other web application techniques lack this graphic support. For implementing the web application in flash, the server side need to be developed in Macromedia's Flex and open source Laszlo framework, and both framework need to be run under Java/Java 2 Enterprise Edition (J2EE) on the server. Moreover, the user's web browser needs to install flash player in order to run the application. The restricted running environment of flash serves as a drawback to this technology, despite its attractive graphic support which will enrich user experience.

3. ActiveX is the name Microsoft has given to a set of "strategic" objectoriented programming technologies and tools. The main technology is the Component Object Model (COM). Used in a network with a directory and additional support, COM becomes the Distributed Component Object Model (DCOM). A component is a self-sufficient program that can be run anywhere in the ActiveX network. This component is known as an ActiveX control. ActiveX is Microsoft's answer to the Java technology from Sun MicroSystems. An ActiveX control is roughly equivalent to a Java applet, and can be reused by many applications. An ActiveX control can be created using one of several languages or development tools, including C++ and Visual Basic, or with scripting tools such as VBScript.

# 3 KooBits System

A system called KooBits has been developed to address those problems mentioned in the previous section. To make the system functional on different platforms, a number of new problems occurred, which need to be appropriately dealt with. In the following sections, issues such as system overview, login system design, KooBits workspace design and KooBits applications implementation will be addressed.

### 3.1 KooBits System Overview

Fig. 1 presents a holistic view of the KooBits system. At the current stage, it consists of the following three major parts:

- 1. KooBits login system: this part manages the user authentication and registration processes. This login system has been designed with very high standard security levels. All the user inputs from the html forms are being filtered and extracted by functions to remove sensitive special symbols. This measure prevents the infiltration of unauthorized person who uses a combination of special sensitive symbols to undermine the database and server side code.
- 2. KooBits user workspace: this web page provides a user interface allowing the users to access the two main functions. First, the user can access to the virtual learning applications under the media authoring system, namely "KooBits electronic book publisher" and "web 2.0 slide show". The user can also access the system database by uploading or downloading various types of media files, such as images, videos and sound.
- 3. KooBits e-book publisher is one of the two main applications being developed under the interactive media authoring system. The application is an effective virtual learning tool for young users to create e-books in a wide variety of categories, such as electronic diary, story book, spelling book, dictionary and even multimedia text book. This software allows the user to merge image, sound, video, 3D animation and text into one user-friendly, coherent format, allowing easy creation of highly interactive electronic content.

### 3.2 Login System Design

To prevent unauthorized access, the system server checks the login credentials against user records stored on a database server. These user records contain information on authorized users to the system. In order to prevent outsider from accessing the content of the site directly, a session has been added into the page. As shown in Fig. 2, the session will verify whether a particular visitor is logged in, and will only display content of the page if he is logged in. The session works on the principle of providing verification with the cookie at the client side computer. When user login, a cookie will be planted in his computer and the browser will be sending the cookie whenever the user is requesting for a new page. The cookie value will then be verified with the system value and data will only be sent if both values agreed.

Visitors will be able to visit the website without logging in. However, there are certain privileges which are only enjoyed by registered users. For visitors, they can see the restricted profiles of current users. In addition, they will be introduced and provided an overview of the whole system. In order to use the application, the users have to log in.

### 3.3 KooBits Workspace Design

The user workplace is an essential part of the system. After successful login, the user will be directed to his workplace, from where the user can manage his files



Fig. 1. System architecture design

or share those files with other users. User can organize their files into folders and put his shared files into shared folders. These files can be sorted or an advance search is incorporated into the application.

The layout of KooBits workspace is shown in Fig. 3. The workspace allows the user to upload media files to the system. Various types of media files such as flash, image, video and animation files are allowed [7]. It also supports a wide variety of file formats such as jpg, gif, wav, wmv etc. After uploading, it allows the files to be integrated into the electronic books created.

User can also create and publish the electronic books in the workspace to be shared within the system. The more books the users published, the more credits will the user earn. These credits can be used to earn further membership into the system or exchange for other services. Users can also add other users as friends in the workspace. This encourages cyber social networking in the workspace helping to build a closely knitted community within the system.

### 3.4 Client-Side Application: KooBits Editor

KooBits Editor allows authors to create next generation multimedia books. Traditionally, authors need to understand various multimedia software applications in order to create a multimedia electronic book. With the introduction of this software, authors now fully concentrate on content creation and focus less on technical aspects. Appropriate images and animations can be added in automatically as author types out stories in text. KooBits client-side application was



Fig. 2. Login system flowchart



Fig. 3. User workspace layout

designed with special considerations to make the user interface straightforward and easy-to-use. The development team had done extensive research targeted at user end by organizing KooBits learning sessions in several primary schools in Singapore, the result shows users as young as 6 years old are able to learn how to use this software within a month. Research has shown that one of the major learning obstacle that young children face is the short attention span, which can be effectively extended by the highly interactive nature of KooBits system [5].

### 3.4.1 Main Features

Table 1 shows a list of functionalities of KooBits:

Features	Description
Text to Speech	This software is able to support both
	chinese and english translation
Artificial Intelligence	This technology enables animation to be
	generated upon text input
Templates	This allows users to create story from
	existing templates
Multimedia	This software allows fusion of different
	media files in one coherent interface
Audio Recording	Audio can be recorded which can be used to
	integrate into the file
Video Display and Export	Video can be displayed and transformed on
	the platform and, the book can be exported
	to video.
Spelling Check	It helps to detect the spelling errors
	committed when creating file

 Table 1. Application functionality list

The client-side application integrates 3D, video, animation, graphics and text into one user-friendly, coherent format, allowing easy creation of interactive ebooks. With the enhanced multimedia technology, KooBits Author further fosters a creative learning experience and allow individuals to express their emotions in different fashions. The system constantly seeks active inputs and activates pictures and animations on the fly. The final output products engage readers with multimedia, animations and Interactive Intelligent Motion whereby pictures and animations move in accordance to reader's interaction. KooBits shortens the creation cycle by letting users begin with formatted templates that they can use to compose various book types such as photo albums, diary entries, comic books and story books etc. Authors need only to focus on creating content. The client side application takes care of the tedious processes such as formatting and page adding before presenting the final product of a complete multimedia e-book ready for showcasing or publishing. Advanced users have the option of having total control over layout and graphics to achieve a more unique presentation. E-books created may be archived on KooBits's online bookshelves, stores and libraries; accumulated works may form part of author's own digital portfolio. They can also be burned into CDs for showcasing and distribution.

KooBits also allows users to easily link words or phrases of the written story to image, video and animation. During auto book playback, the text will appear gradually and triggers digital files to appear as story progresses. This feature engages the child in the writing and creation. "KooBits can be used to create comics, video books, storybooks, album, diaries, instructional text, teaching books, excursion recounts, science experiment reports, and many other book types" [12]. The content can be a mixture of flash, 3D, animations, videos, text and graphics. It is able to support English, Mandarin, Tamil, Malay and any other Unicode inputs. KooBits transforms writing from a passive subject into



Fig. 4. KooBits reading mode interface

a fascinating learning adventure for users. The created KooBits interactive ebooks allow active input from readers and reward them with unique animated e-books to call their own. It encourages self-expression and heightens authors' propensity for independent work.

# 3.4.2 User Interface

KooBits is a user-oriented software application which places heavy emphasis on user experience. The graphical user interface of the software is designed based on the conventional design heuristics [10]. Firstly, it is taken into account that the electronic books editor is a novel concept and users may not get use to it initially. Hence, play, fast forward and stop buttons are imported from conventional music player to give user a sense of familiarity. Studies have also been made into commercial software applications to ensure that the presentation of user options follow industrial standards. This makes the software more user-friendly. Fig. 4 shows the KooBits user interface design.

The electronic book presentation is designed to simulate that of physical book and physical concepts such as flipping is emulated in the virtual space. With these features, users are able to relate their physical world experience into the virtual space thereby enhancing user experience. To open previous works, Fig. 5 indicates that the software provides a visualization using a bookshelf as a metaphor. Unlike conventional software, organizing books on the bookshelf allows better categorization. It also significantly improves software interface presentation by allowing the user to relate their daily experiences onto the virtual space by making use to metaphorical representations.

# 3.4.3 Client-Side Application Implementation

KooBits integrates 3D, video, audio, animation, flash, images and text all into one user-friendly, coherent format, through its advanced "Media Fusion" technology



Fig. 5. KooBits book shelf interface

[24]. It shortens the content creation cycle by letting users import a wide range of digital formats under one editing environment, and facilitates the manipulation and editing of highly interactive content in real-time.

This software allows authors to publish next generation multimedia books. In the past, authors need to understand various multimedia software applications in order to create a multimedia electronic book. With the introduction of this software, authors now fully concentrate on content creation and focus less on technical aspects. Appropriate images and animations can be added in automatically as author types out stories in text.

Content-driven animation is another technical advance of this software application. Instead of using key frames to create animation, users of KooBits can use storyline to generate animations and trigger dynamic actions make up of various digital formats. This significantly shortens the process of creating an interactive media from hours to minutes. With embedded AI components, KooBits also allows users to create animation content based on user's text input. For instance, a fox avatar will be generated when the word "fox" is keyed in. It is primarily designed for users to aid them in expressing their creativity. This technology is redesigned to aid users in learning abstract concepts which are better explained pictorially. Developing an animations generator based on text input application, therefore, allows user to interact with the system. In this application, the system will generate the conceptual animation to aid users to visualize the conceptual objects. It also permits interaction by responding to users' text input to adjust the animation accordingly.

KooBits helps to create digital books in a fun and interesting way via its powerful built-in artificial intelligence system. This Artificial Intelligence system constantly seeks active user inputs and responds with picture and animation generation. It resulted in a final output - electronic books that engage readers with multimedia and animations coupled with real-time interactive intelligent motion. This electronic book can be used for playback with the story slowly unfold.

#### 1) KooBits Text to Speech Converter

Another key technology of KooBits is its sophisticated text to voice converter which transforms user's text input into speech effortlessly. The current version supports all languages. This text-to-speech system is developed based on a text normalization component. The input text is converted into a series of words. Text normalization will isolate the words in the input text. The system then looks for significant symbolic representations to analysis. It will then be converted to words. If abbreviation is encountered, the normalizer will compare the abbreviation with the database to look up for the appropriate word. After the word is normalised, it is then passed on to the homograph disambiguation stage. This stage analysis the most appropriate pronunciation based on the context of the input text. After the disambiguation process, the information will be passed onto the pronunciation module. The pronunciation module will first look out for the appropriate word in the system pronunciation lexicon. If the word is not found in the lexicon, the engine will pronounce it using the letter to sound rule. With that, the appropriate sound will be determined and output to the user. Using this feature greatly enhances interactivity which significantly improves user experience.

### 2) KooBits Text Parser

In order to understand the user's input and react to it by generating various animations. A text parser has been designed to filter the user's input and feed the filtered result into the application core modules. The text parser takes typed input from the user and simplifies it to something that KooBits can understand. Normally, words with similar meaning are converted into the same word, for example, the words "acquire" and "obtain" are filtered by the parser and classified into the same verb that means get. A database has been built up that relates different words and phrases with similar meanings. The KooBits Text Parser makes it much easier for the client side application to react on user's input. After text filtering, the system core modules does not have to check whether the command is "obtain the gem", "take the gem", "take gem", "get gem", "discover the precious gem", and so on, because the KooBits Text Parser has already stripped these various phrases down to something like "take gem".

For the users, the client-side KooBits application is more flexible. The users do not have to input exactly the right words, because the application can understand the same word or phrase that appears in a few different forms.

## 3.4.4 Applications of KooBits Editor

KooBits offers a high degree of freedom to customize, it can be used to create comics, video books, storybooks, photo albums, diaries, instructional text, excursion recounts, even science experiment reports. The content can be a mixture of flash, 3D, animations, videos, text and graphics. The software is able to support English, Mandarin, Tamil, Malay and any other Unicode inputs.



Fig. 6. Diary book



Fig. 7. Spelling book

As shown in Fig. 6, this software can be used as an interactive diary or blog for users. The current form of diary is either in the form of diary book or website blog. This will help to meet the market demand for the next generation electronic diary.

An extension of this technology can also be used to create an educational tool to helping user to learn spelling. This can be used as an educational for users to learn spelling. Assignment will be given to the user via the electronic books. Hints will be given to the users for them to key in the correct words. Upon keying



Fig. 8. Grammer exercise book

in, the software will feedback to the users on the results. If the result is correct, a relevant descriptive animation of the spelling word will appear on the interface. This helps the user to enforce what was learned. Fig.7 shown the spelling book.

KooBits can also be used as a language training book. The software is very powerful and efficient in detecting the correct grammar of the language. Thus, it can help the user to master a language on his own via the interactive feedback. Appropriate animation will be shown to enforce the learning when the user keys in the right grammar. When the grammar is wrong, the software application will feedback to the user the area he needs to improve on. This grammar exercise book is shown in Fig. 8.

It can also be used as an interactive dictionary. Traditional dictionary allows one to look up for the meaning of the words and detail description. In recent times, dictionary has moved online and people can easily search words via website. However, explanation of the words remains quite static. Using KooBits, a dynamic dictionary can be created. After the user keying in the words, the software application can provide the meaning of the words. In addition, relevant images and animations can be used to explain the meaning of the word much more effectively, thus speeding up the learning process substantially.

### 3.5 Server-Side Application: KooBits Publisher

KooBits Publishing Network is an online platform for all the users to publish their electronic books and also view the works done by other users. Within this website, all the registered users can upload the completed piece of works to the database and make it available for all the other users. "The book authors can also have the option to source, develop, share or market their e-books" [12]. Besides



Fig. 9. KooBits publishing platform

uploading and downloading e-books, the users can also upload media files like photos, pictures, drawings, storylines, animations, music and video to co-develop and co-publish content with other users. Fig. 9 shows the layout of the KooBits publishing interface.

Registered users in this networks are able to earn Koobits credits through active participation of publishing activities. These Koobits credits in turn can be exchanged for free membership and other members benefits. In this way, users are being encouraged to publish more which results in large amount of user generated content. This enhances the value of the network.

# 4 WEB 2.0 Media Publisher

KooBits was primarily developed as a client based solution to address the interactivity issue. The industry has witnessed the migration of media authoring development to the World Wide Web in the recent years [1]. Web 2.0 Media Publisher extended the KooBits media authoring system and is developed as a server based alternative. This web based application allows the user to create multimedia documents. The documents created are being wrote into xml format [20]. This publisher is built upon the foundation of Tan and Wu [27, 29]. These user created documents are stored in the system database and can be shared concurrently accessed by multiple users [24].

The client program in the last generation of web applications will send a request and load a new page from the server only when the user clicks on a button to trigger an event. The asynchronous HTTP call is different from the traditional http call in various aspects. Firstly, the http request is being processed in the background such that no refresh of the webpage will be created. The client-side program will only take the returned data to update the existing web page. Secondly, the asynchronous request allows the user to perform other actions before the reply is received from the server. As a result, it minimizes the user's waiting time for process completion and allows multiple requests to be made from the client.

### 4.1 Web 2.0 Slides Show

Web 2.0 slides show application is functionally similar to the desktop application - Microsoft PowerPoint. At the client side, HTML and CSS are used to create the graphical user interface, and JavaScript code controls the logic. The PHP scripts at the server side is responsible for handling request from function calls, and MySQL is used to store the content.

To make the system easier to develop and maintain, we have taken a modular approach to break down the control logic into different functional modules, with each of them perform certain specific tasks and provide functions that can be called by other modules. All these modules are implemented as different JavaScript (.js) files. Fig. 11 shows the structure of the application.

Every slide consists of a number of editable rectangular boxes. These boxes could be a text box, or an empty box containing picture or video. The edit box



Fig. 10. Web 2.0 slides show structural diagram

is created and handled by the EditBox module, and each of them is editable, resizable and draggable. The edit box module is defined in the file editBox.js, in which there are two classes: editBox and editBoxBuilder. The class editBox is used to construct the static edit box, while the class editBoxBuilder is used to add events to the edit box. To create an edit box, the main module will call the editBoxBuilder class, which will in turn call the editBox class to create a static box, then adding events to it and return the final edit box to the main routine. The editBox class is used to create a static edit box. An edit box is created as a HTML div element with visible border. In this edit box, certain properties of the edit box will be kept track, including its x-position, y-position, width, length, state (idle, focus, editing). The contents of a edit box is empty by default, but it can also be used to contains contents like text, picture, video or geometric shapes.

The Slide module defined in JavaScript file slide.js is use to create a new slide. It is able to create a default empty slide with no content, or open the previously saved slide and use the stored content to create the new slide. Similar to the structure of edit box, the slide module has two classes defined. The slide class is used to build a static slide. The static slide is build by creating a HTML div element as the slide, and calls the editBoxBuilder class to build the various edit box inside the slide. The slideBuilder class is called by the main program to build the actual slide. The slideBuilder class calls the slide class to create a static slide, and adds the event functions to the slide so that it could respond to different user action. The addEditBox() function in the module is used to insert extra edit box into the slide. Hence multiple edit boxes could be used in the same slide to contains various content such as text, picture and video. To insert a new text box into the slide, click on the Insert button on the menu, then choose New Text Box. A new text box will be added to the current slide.

The slide show module is used for creating the slide show effects in the slide presentation. This module makes use of the JavaScript library provided by Script.aculo.us, in which effects are implemented as functions. When an effect function is called with an object as a parameter, then the effect will be applied to the input object. For example, if the slide show module calls Fade(slide1), then the effect fade is applied on object slide1, and the slide will become more and more transparent and disappear eventually.

The slide show can be started by clicking on the slide show icon in the tool bar. A new window will appear to display the slide show. There are many slide show effects available, such as fade, slide down/up, blind down/up, grow, fly in/out. However, the various effects are a combination of a few numbers of simple core effects, including:

- Change of opacity reduce the opacity of an object to make it more transparent;
- Translation move an object in the left/right or up/down direction;
- Resizing increase or decrease the size of an object, including all of its elements

These three fundamental effects are the building block of the many slide show effects implemented at the later stage. With the combination of the above core effects, plus time control, many Microsoft PowerPoint slide effects can be produced. For example, we can reduce the opacity of an object by certain percentage after a few millisecond to produce the fade effect; or we can make a small size slide to move in a spiral path with expanding size until it reaches the final full screen position, by increasing the slide size and changing its translation direction every few millisecond. This module runs on top of the Communication module and it is used to save the contents of the slide on the server. The Save function can be accessed by clicking on the Save To Server menu.

The dictionary module is a JavaScript widget added to the system. It allows the user to enter a word to search for its meaning. The searching of the definition of the key word is actually done on the server, by making use of some other online dictionary database. When a user enter a key word and click search, the client program will initialize an asynchronous HTTP call to the server, with the search key word as input. After receiving the word, the server side will call the dictionary handling function to connect to some online dictionary websites and do the searching. As these websites will return a web page containing the definition of the word together with other content like advertisement, the server side handling function will parse the result and filter out the irrelevant content. Afterwards, the function will reply the definition of the word to the client program, which will display the definition in the JavaScript widget. Fig. 12 shows the dictionary interface.

The dialog box consists of three parts: one is the text input field in which user can enter the word to be searched; another is the search button; the third one is a text area in which the meaning of the word will be displayed. In this module, two methods of searching are implemented. The first one is the normal search. User opens the dialog box and type the word in the input text field. After clicking "Search", the definition of the word will be displayed in the main text box, after the program receives the reply from the server.



Fig. 11. Dictionary javascript widget

Another method of searching is the highlight search. In this way the user only need to highlight the work he want to look up, then open the dictionary dialog box following the same steps described above. The dictionary module will automatically check for highlighted words in the slide. If there is any highlighted word, it will initialize the search and display the result once it is obtained from the server.

It is also possible to implement the dictionary function by installing a dictionary database on the server side, and have the advantage of fast searching and less overhead since there is no content filtering. In , we use available online dictionary websites for the advantage of larger database and more information, without the need of database maintenance and update. It is a suggested practice in web application development, to make use of available online resource as much as possible. This technique could be further enhanced to connect the server to other powerful internet search engine, so it can perform various kind of searching.

User can add in geometric shapes, such as square, triangle, circle, star and arrow in the slide. These shapes are generated using the shape module, and then inserted into an editable box so that it can be moved and resized.

Although the system only provides predefined shapes for user, shape module is function that can be used to draw any arbitrary geometric shape. This module takes a point array as the function parameter. This point array is an array that contains the coordinates of all the vertexes, with the first array item as the coordinate of the first vertex (starting point). The function will start from the first vertex, then draw connecting line to the second vertex, and so on. When it reaches the last vertex, the function will automatically connect it to the first vertex to draw a closed shape. Vector Markup Language (VML) is used to draw the various shapes in the Shape module. VML is an application of Extensible Markup Language (XML) 1.0 which defines a format for the encoding of vector information together with additional markup to describe how that information may be displayed and edited. It supports the markup of vector graphic information. Within VML the content is composed of paths described using connected lines and curves. The markup gives semantic and presentation information for the paths.

Besides being resizable and draggable, the geometric shape created by the Shape module is able to rotate. When user double click the edit box that contains the geometric shape, the edit box will be in "edit" state, in which user could drag the shape to rotate it. The shape class defined in the module is the one that creates the geometric shape. The rotate() function in the class is used to produce the rotation of the geometric shape. The rotate() function is called when the mousemove and mousebuttondown event occurs in the edit box "edit" state. The function will detect the co-ordinate of the mouse and the one of the center of the shape. By comparing these two sets of co-ordinate, the function will decide the direction (clockwise or anti-clockwise) and turns the shape by a small degree. Rotation will continue as long as there is mouse movement and the mouse button is being pressed.

To access the upload picture function, click on the Upload Image menu. A dialog box will appears and allow user to browse the file in his local computer. After selection the user can click on the Upload button to transfer the file to the server. The Gallery module is developed to be used together with the Upload Picture module. This module is used to allow user to brows the image contents saved in the server, and the user could select and insert the picture into the slide.

To open the gallery function, click on the Gallery menu. A dialog box will appears in the workspace. In the Gallery dialog box, there is a panel which displays all the file names of the image files saved on the server. User is able to select the file in the panel, and the associated picture will be displayed in the Preview panel. After finding the desired image, user could click on the Insert to add the picture into the slide. At the current stage, the gallery is connected to the server and to view the contents in the user's folder.

### 4.2 Communication Modules

In order to allow users to communicate and collaborate on the media authoring system, both text and voice based communication modes are made available for the users to carry out discussion. Text-based communication is carried out in traditional chat-room style while voice-based communication allows the users to speak using desktop microphones. This combination creates a communication method that is highly interactive and at the same time less demanding in terms of bandwidth compare to other methods like video conferencing and video broadcasting.

#### 4.2.1 Text-Based Communication

Users within can communicate with each other through text-based communication. This communication is conducted for users to interact with each other in real time. As an educational system, allowing users to interact with each other is an important feature as it allows ideas to be shared. Users can also clarify their doubts through this form of peer support.

## 4.2.2 Voice-Based Communication

In comparison, audio-based communication is more complicated due to its higher requirement in terms of bandwidth. As the transmission of audio data can generate significant amount of network traffic, measures should be taken to reduce the strain on the network. The interactive learning system conserves bandwidth through the use of audio compression techniques and exercises control over the audio transmission.

In our audio transmission module, Audio data stored in Pulse Code Modulation (PCM) format is compressed by GSM 6.10 audio codec, which is capable of achieving a transmission rate of 8-16 kbps. After compression, the compressed audio data is then sent across the internet to the all the listeners. Upon arriving at a listener's end of the connection, the client application converts the compressed audio data back to the PCM format and plays it through the PC speakers. Using this method, the audio communication process does not involve the transfer of large audio files, but instead it relies on the streaming of audio data packets. Thus, the network traffic can be lowered significantly.

### 4.2.3 Concurrent File Editing

In this system, users can collaborate with each other to work on group projects by working on the same documents concurrently [18, 22]. All the users are given the permission to create projects in the workspace. After creating the project, the creator can add his group mates into the project group and declares certain documents as shared files, enabling it to be accessed and edited by all the other group members. At the same time, the user database will be updated and these shared documents will appear inside the workplace of all other users in the project group.

In order to allow more than one user to edit a document concurrently, all the users should be mutually excluded from accessing the particular part of the content that the users are currently working on. This is to prevent problems from arising when one user overwrites the content of the same section which another user is still working on. Thus, mutual exclusion is necessary to protect the integrity of the data. The mutual exclusion rule in this system is such that when the first user accesses a certain section of a document, this section is blocked from all the other users until this user finishes editing and releases this section. For example, Fig. 13 shows a user's view when he is doing concurrent editing of an online power point document. The title text box appears in gray color indicating that this text box is currently being edited by another user.

The concurrent file editing feature is implemented through a few critical steps.

- 1. A module is written to send a signal to the server. The signal serves as an indication to the server should a user decides to edit a certain section of a document.
- 2. When the server receives this signal it will then update other clients who are working on the same document to reserve the textbox that the editor is currently working on.
- 3. The user is granted unique access to the selected section and starts updating data to the server using ajax communication model.

4. On the server side, all the updates from users are entered into a queue. The server periodically reads from the queue and sends requests to update all the documents stored on the user's computers. Fig. 13 shows how this mechanism works.



Fig. 12. Concurrent file editing

### 4.3 Future Development

The system can be further developed and evolve into a virtual online learning platform by integrating several different modules introduced in the above sections: text/voice communication, concurrent file editing and Web 2.0 Slides Show [23]. This will become a more interactive solution over existing technology which primarily promote one way communication [16, 17, 19]. In fact, developing this technology will enhance the effectiveness of distance learning [6, 15]. Fig. 14 illustrates how such a virtual online lecture network can be built and how the lecture can be carried out.

Only lecturers are assigned the permission to initiate new virtual lectures. An online power point document will be prepared in advance and attached to that lecture session. Under this system, it brings users from different regions to logon to the system and attend the lesson together. During each virtual lecture session, all the users will see the same online power point document on their client machine and receives voice communication from the lecturer in a way similar to broadcasting. However, a significant difference from conventional broadcasting is that all the users are allowed to signal to the lecturer by "virtual hand rising" [25]. If a user wants to interrupt the session to ask a question, an option button is provided to signal this intention to the lecturer. Upon receiving the request, the user can talk to the whole lecture room if the lecturer granted the permission to speak up via an accept request button provided. The users can then choose to



Fig. 13. Online virtual lecture

use either text or voice-based communication mode according to the bandwidth limitation of his/her internet connection.

Besides text/voice discussion, the lecturer can ask users to solve problems using the concurrent file editing feature. On the web 2.0 slides show system, each user can be assigned to work on a mathematics problem and write their answers in separate text boxes. They can look at each other's solution, compare and give comments, thus enhancing interactivity in learning.

At the end of each virtual lecture, all text-based communication and problem solving sessions will be automatically recorded in terms of history log on the system server. This can be accessed by users again for review purposes.

# 5 Conclusion

This paper discussed two different systems, KooBits Media Authoring System and Web 2.0 Media Publisher, which enhance the effectiveness of learning process. These two systems incorporate creativity, innovation and cutting edge technologies to facilitate this process. These systems promote interactivity which is a critical factor in effective learning. KooBits Media Authoring System is developed based on client based technology which requires installation. Web 2.0 Media Publisher build on the idea of KooBits and is developed into a Web 2.0 system. This enables the system to be more widely accessible and free the user from installation. Traditional client based applications have slowly developed into server based web applications. It is anticipated that this shift from client application to the server side is likely to continue in the future.

Acknowledgements. The authors thank the inputs from Xiaoyan Stanley Han and Xiangdong Chen, Personal e-Motion Pte Ltd.

# References

- Mark, A.K.: Integrating open hypermedia systems with the World Wide Web. In: Proceedings of the Eighth ACM International Hypertext Conference, Southampton, UK, pp. 157–166 (1997)
- [2] Bailey, B., Konstan, J., Cooley, R., Dejong, M.: Nsync A Toolkit for Building Interactive Multimedia Presentations. In: Proceedings ACM Multimedia1998, Bristol, UK, pp. 257–266 (1998)
- [3] Knightly, E., Templin, F., Banerjea, A., Mah, B., Zhang, H., Ferrari, D.: The Tenet Real-time Protocol Suite: A Demonstration. In: Proceedings of the second ACM international conference on Multimedia, San Francisco, California, United States, pp. 489–490 (1994)
- [4] Seongbae, E., No, E.S., Kim, H.C., Yoon, H., Maeng, S.R.: Eventor: An Authoring System for Interactive Multimedia Applications. Multimedia Systems 2(3), 129– 140 (1994)
- [5] Essa, E.: A Practical Guide to Solving Preschool Behavior Problems, pp. 382–393 (2001)
- [6] Sébastien, G.: Contextualizing Discussions in Distance Learning Systems. In: Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004), Joensuu, Finland, August 30-September 1, 2004, pp. 226– 230 (2004)
- [7] Cruz, G., Hill, R.: Capturing and playing multimedia events with STREAMS. In: Proceedings of the second ACM international conference on Multimedia, San Francisco, California, USA, p. 466 (1994)
- [8] Hamakawa, R., Rekimoto, J.: Object composition and playback models for handling multimedia data. Multimedia Systems. In: Proceedings of the first ACM international conference on Multimedia, Anaheim, California, USA, pp. 273–281 (1993)
- Hardman, L., Rossum, G., Bulterman, D.: Structured Multimedia Authoring. ACM Transactions on Multimedia Computing, Communications and Applications 1(1), 89–109 (2005)
- [10] Nielsen, J.: http://www.useit.com/papers/heuristic/heuristic\_list.html
- [11] Wei, J.: Web page based dynamic book for document presentation and operation. US 0030719 A1 (2004)
- [12] KooBits web site, http://www.KooBits.com
- [13] Soares, L.F., Rodrigues, R., Saade, D.M.: Modeling, authoring and formatting hypermedia documents in the HyperProp system. Multimedia Systems 8(2), 118– 134 (2000)
- [14] Hardman, L., Bulterman, D., Rossum, G.: The Amsterdam hypermedia model: adding time and context to the Dexter model. Communications of the ACM archive 37(2), 50–62 (1994)

- [15] Castro, M., Pérez-Molina, C.M., Colmenar, A., Mora, C., Yeves, F., Carpio, J., Peire, J., Daniel, J.: Examples of Distance Learning Projects in the European Community. IEEE Transactions on Education 44(4), 406–411 (2001)
- [16] Microsoft's NetShow, http://www.microsoft.com/netshow
- [17] Microsoft's Vxtreme, http://www.microsoft.com/netshow/vxtreme
- [18] Streitz, N., Haake, J., Hannemann, J., Lemke, A., Schuler, W., Schtitt, H., Thuring, M.: SEPIA: A Cooperative Hypermedia Authoring Environment. In: Proceedings of the 4th ACM Hypertext Conference, Milan, Italy, p. 1122 (1992)
- [19] Real networks, http://www.real.com
- [20] Andrews, R.L., DeFilippis, J.M., Murphy, D.S.: Creating an HTML document from a source document. US 019489 (2003)
- [21] Giachetti, R.E.: Integrating hypermedia design concepts with a systems analysis and design methodology to develop manufacturing web applications. Int. J. Computer Integrated Manufacturing 18(4), 329–340 (2005)
- [22] Costa, R.M., Moreno, M.F., Rodrigues, R.F., Soares, L.F.: Live Editing of Hypermedia Documents. In: Proceedings of the 2006 ACM symposium on Document engineering, pp. 165–172 (2006)
- [23] Benjamin, R., Davenport, G.: Structured content modeling for cinematic information. ACM SIGCHI Bulletin 21(2), 78–79 (1989)
- [24] Deshpande, S., Hwang, J.: A Real-Time Interactive Virtual Classroom Multimedia Distance Learning System. IEEE Transactions on Multimedia, 432–444 (2001)
- [25] Ge, S.S., Tok, M.Y.: Enhancing Online Education Using Collaboration Solutions. Journal of Educational Technology Systems 31(4), 361–380 (2003)
- [26] Antonio, S., Fox, E.A., Hix, D.: The Integrator: A Prototype for Flexible Development of Interactive Digital Multimedia Applications. Interactive Multimedia 2(3), 5–26 (1991)
- [27] Tan, C.K.: Web-based Desktop Application. B.Eng Final Year Project Thesis, ECE department, Faculty of Engineering, National University of Singapore (2002)
- [28] Thuong, T.T., Roisint, C.: A Multimedia Model Based on Structured Media and Sub- Elements for Complex Multimedia Authoring and Presentation. International Journal of Software Engineering and Knowledge Engineering 12(5), 473–500 (2002)
- [29] Wu, Z.: WEB 2.0 Slide Show Presentation System. B.Eng Final Year Project Thesis, ECE department, Faculty of Engineering, National University of Singapore (2007)
- [30] Nicole, Y., Meyrowitz, N., Dam, A.: Reading and writing the electronic book. IEEE Computer 18(10), 15–30 (1985)