Informatics in Education, 2005, Vol. 4, No. 1, 5–18
© 2005 *Institute of Mathematics and Informatics, Vilnius*

Development of Multimedia Authoring Tool for Educational Material Disseminations

Mohamad Nizam AYUB, Santhimathy T. VENUGOPAL,
Nurul Fazmidar Mohd NOR

Faculty of Computer Science and Information Technology
University of Malaya, 50603 Kuala Lumpur, Malaysia
e-mail: nizam_ayub@um.edu.my, santhimathy@um.edu.my, fazmidar@um.edu.my

Received: November 2004

Abstract. The future of multimedia technology is guided on how it is utilized. One particular domain that has shown a lot of potential form multimedia use is the education sector. This paper will try to explain the process of development a multimedia tool that will be used to construct educational teaching and learning materials or otherwise known as courseware. The development of this authoring tool will provide the basis for identifying practices that can be use of core multimedia technologies for supplementing educational resources.

Key words: multimedia learning, authoring tool.

Introduction

Authoring tools and multimedia-oriented applications have been quite popular since the mid 1990s. The ability of authoring systems to integrate numerous multimedia elements such as audio, video, animation, text, pictures and images, into a single application or presentation has brought tremendous changes to the way society deals with and disperses information. In addition, the ability to inject interactivity whereby, end users can dynamically engage in direct interaction with the application and navigate the movements or interaction with the multimedia application has brought a huge impact on the educational and research field in general.

It has been noted and observed in the educational field in particular, that the more variety of resources used in the learning process, the better the ability to heighten or enhance the human capability to absorb and retain facts of the learning material. This is due to the simultaneous use of the auditory and visual senses combined with direct real-time interaction with the learning components. This paper will focus on the development of an authoring tool that will assist educators in their preparation of teaching and training material. At the end of this paper, the author will suggest some features compulsory in order to develop good authoring systems, which could enhance the educator’s productivity and student’s understanding.

About Authoring Systems

An authoring tool can be defined as "an application development environment for non-programmers". However, this does not specifically define those only tools, which require no coding from the author can be deemed to "authoring tools". Some existing popular tools such as Toolbook and Guide use their own programming tools called scripting. The only difference between such tools with a programming language is that a non-programmer can learn to use such a tool in a short space of time (relative to that taken to learn a conventional programming language, such as C++ or Pascal) and use it to produce high-quality applications.

An authoring system is a program which has pre-programmed elements for the development of interactive multimedia software titles. Authoring systems vary widely in orientation, capabilities, and learning curve. There is no such thing (at this time) as a completely point-and-click automated authoring system; some knowledge of heuristic thinking and algorithm design is necessary. Actually authoring is just a speeded-up form of programming without the need to know the intricacies of a programming language but an understanding of how programs work is necessary (Merrill, 1985).

Design Metaphor/Authoring Paradigm

The following are methodologies by which the authoring system accomplishes its task. Many authoring systems are considered to be based around one of the following designs:

- *Scripting Paradigm*
  The scripting paradigm is the authoring method closest in form to traditional programming. Scripting is programming language, which specifies multimedia elements, sequencing, hotspots, synchronization, and many others. Scripting languages do vary; in terms of how much the language is object-oriented or object-based. Although the scripting paradigm tends to be longer in development time (it takes longer to code an individual interaction), it offers more interactivity. Since most scripting languages are interpreted, instead of compiled, the runtime speed gains over other authoring methods are minimal. The media handling can vary widely.
- *Card-Based/Scripting Paradigm*
  The card/scripting paradigm provides a great deal of power (via the incorporated scripting language) but suffers from the index-card structure. It is excellently suited for Hypertext applications, and supremely suited for navigation intensive applications. Many entertainment applications are prototyped in a card/scripting system prior to complied-language coding. The best-known card based tools are Hypercard for MAC and Toolbook for Windows. An application generated using a card-based system is made up of one or more stacks of cards, or pages, navigated using paging or hypertext links. They are particularly well suited to produce hypertext documents.

- *Icon-Based/Flow Control Paradigm*
  The most well known example of this type is Authorware, which is available for the MAC and IBM PC compatible machines. In this type of program, icons are selected and used to generate a flow chart, which controls the application. In Authorware, icons include display icons, where text and graphics can be added, decision and branching icons, animation and sound icons and so on. Although these types of programs can often be used to generate simple applications very quickly, more complex applications require much more skill and planning. This tends to be the speediest (in development time) authoring style, which is best, suited for rapid prototyping and short-development time projects. Many of these tools are also optimised for developing Computer Based Training (CBT). The code of the paradigm is the Icon Palette, containing the possible functions/interactions of a program, and the Flow Line, which shows the actual links between the icons.
- *Frame Paradigm*
  The frame paradigm is similar to the Iconic/Flow Control paradigm in that it usually incorporates an icon palette; however, the links drawn between icons are conceptual and do not always represent the actual flow of the program. This is a very fast development system. The best of these have bundled compiled-language scripting, such as Quest (whose scripting language is C) or Apple Media Kit.
- *Cast/Score/Script Paradigm*
  The cast/score/scripting paradigm uses a music score as its primary authoring metaphor. The synchronous elements are shown in various horizontal “tracks” with simultaneity shown via the vertical columns. The true power of this metaphor lies in the ability to script the behaviour of each of the cast members. The most popular member of this paradigm is Macromedia Director, which is used in the creation of many commercial applications. These programs are best suited for animation-intensive or synchronized media applications.
- *Hierarchical Object Paradigm*
  The hierarchical object paradigm uses an object metaphor (like OOP), which is visually represented by embedded objects and iconic properties. Although the learning curve is non-trivial, the visual representation of objects can make very complicated constructions possible.
- *Hypermedia Linkage Paradigm*
  The hypermedia linkage paradigm I similar to the frame paradigm in that it shows conceptual links between elements; however, it lacks the frame paradigm’s visual linkage metaphor.
- *Tagging Paradigm*
  The tagging paradigm uses tags in text files (for instance, SGML/HTML, VRML, 3DML, and Winhelp) to link pages, provide interactivity and integrate multimedia elements.
- *Time-Based Paradigm*
  This kind of tool often uses a visual time line on which multimedia events are placed to control the flow of the application. Examples include Macromedia’s Di-

rector and Action! Simple time based tools are often used to generate multimedia presentations.

Authoring Interface

The interface of an authoring tool plays an important role in attracting the users to use the tool. An appropriate graphical user interface should be selected for different multimedia applications, which naturally have different roles to extend. In the authoring environment, the authoring interface is referred to as the authoring metaphor. This basically means the way applications are structured. Some of the current examples that are available are given below.

Slide-show Metaphor

Each screen of a presentation is seen as a slide object. This metaphor is an excellent choice for authoring linear presentations. However, it is not suitable for random interactivity. Some examples of current tools that apply this metaphor are Microsoft's Powerpoint and Lotus's Freelance.

Book Metaphor

In this metaphor, the application is thought of as a book containing many numbers of pages, which can contain any number of multimedia objects and interactivity. Each object has an associated script, which contains any event handler. It is suitable for Encyclopedia, fairy tale story, etc. It also requires programming skills of a certain level. An example of such a tool is the Asymetrix's Toolbook.

Timeline Metaphor

A timeline metaphor works with a time scale diagram and places objects and events on the time scale in correct relationship. It is especially valuable when dealing with a number of dynamic activities. Some authoring tools that apply this metaphor are Macromedia Director and Action!

Icon Metaphor

The application structure is built by dragging icons from an icon palette into the workplace to form a flow-line of icons. These icons represent objects and events that will occur according to the sequence of the arrangement of the icons. Authors can now concentrate on the application logic and content rather than on the details of programming. Developing sophisticated application is still a difficult and complex task. For example, AimTech's IconAuthor and Macromedia Authorware.

Courseware

Courseware is defined as instructional material in an interactive mode to facilitate and control the individualized learning environment for students (Vassileva, 1992). It can be used to provide instructional material to a group of students or for individual students.

Courseware can be further subdivided into a few instructional methods such as, drill-and-practice, tutorials, and problem solving. These programs supplement or enrich the learning situation whereby the success of developing the courseware is governed by three major factors:

- the content and pedagogical quality of the learning materials;
- the motivational quality of the materials;
- the amount and character of faculty support in the overall learning situation.

Simplistic course ware can be developed by using easy-to-learn programming languages as well as utilizing minimal hardware support by leading to limited results (Young, 1984). Complex courseware on the other hand, requires authoring tools or systems, which permits such features as the usage of text and graphics editors, and record-keeping procedures as well as large-scale computing capability to support such use. With the availability of cheap, but powerful microcomputers, which were one of the major drawbacks of former computer-aided instructional material attempts, namely high cost, has gradually disappeared (Woodhouse and McDougall, 1986). Its now being widely used throughput various level of education, as a teaching tool in lieu of the more traditional methods of lecturing, working with programmed texts or using diverse types of audio-visual instructional methods.

Course Tutorial

This approach is one of the earliest and basic uses of computers in the instructional process. It implies a series of four steps in the teaching of any given concept: (i) stating the item of knowledge that is to be taught; (ii) clarifying and elaborating the item in a series of sentences; (iii) restating the item to improve understanding; and (iv) making a transition to the next concept or item to be taught (Jona, 1991). This simple four-step model of learning process could be implemented in a variety of ways, such as the early use of the computer as an automatic page-turning device that merely produced what was on the printed page, the later, the employment of the computer's strength of logical branching. The ability to branch to an appropriate set of material depending upon the student's current state of knowledge is an ability that is uniquely provided by the logical capability of computers.

Drill-and-Practice

Drill-and-practice is another author-controlled approach to developing courseware. In such approach, the aim is to take over the main responsibility for developing the student's skill in the use of a given concept. This involves leading the student through a series of examples where he can practice the material he has already learned or can have it repeated for him. The assumption with a drill-and-practice system is that the student has already had the concept presented to him, that the material has been seen before, and the purpose now is to gain familiarity and develop some familiarity with the ideas. Since the purpose of the drill-and-practice is to increase learning effectiveness through repeated practice based on a stimulus-response (S-R) theory of behaviorism, the frequency of repetition should have a direct effect on achievement (Skinner, 1986). The results of the study suggest that drill-and-practice-based lessons can be an effective means of teaching students of varying learning styles.

Problem-Solving

The problem-solving approach to courseware is based on the hypothesis that the thought required to write a computer program aids the development of general problem-solving skills. It differs from simulation in that the activity of programming, rather that the specific subject of the program, is considered central. In simulation, the student is given the programmed model to use, not asked to write or change this program. The underlying educational philosophy of the problem solving is a belief in what may be summarized as “learning by doing”, as opposed to simulation’s “learning by seeing”. Problem solving is an important high-level skill and regarded by some to be the most complex of learning activities (Gagne, 1977). In general, students are expected to develop these by some ill-defined process involving illustration and performance. The role that programming might play in this is that computers enable to develop, investigate and experiment with their own ideas and theories in relatively enjoyable and informative fashion.

Advantages and Disadvantages of Courseware

The effectiveness of courseware is difficult to define. Some possible criteria for measuring its effectiveness are amount and retention of learning, attitudes to the topic of study in general, and attitude to computers. In general, well-designed and tightly controlled evaluative studies of the use and effectiveness of courseware in the learning process are rare. However, from several studies that have been conducted, some generalized conclusions can be drawn as to the advantages to the usage of courseware in an educational system:

- The courseware’s ability to individualize the instructional process (Majidson, 1978). It individualizes instruction by responding appropriately to current student input, possibly taking into account also the performance of the student. This could extend to the presentation of alternative materials or methods. It also allows the individual student to have some control over the content, sequence, and pace of his/her content.
- Courseware with multimedia contents involves students actively in the learning process. It is impossible for the student to be a totally passive member of the situation, and this activity and involvement facilitate learning (Chamber and Sprecher, 1980).
- Partly as a result of interactive student participation, the computer and courseware offers fast feedback. Students are kept informed of their progress through immediate feedback, presentation and achievement summaries (Kulik and Kulik, 1985).
- Instruction is systematically prepared, sequenced, tested, and revised. Reinforcement of learning is immediate and systematized. Students are allowed to review previous instruction, request special help, or to continue on to enrichment activities.
- Courseware frees teachers for other necessary work, and hence increases educational productivity. Teachers are able to devote more time to the personal, human considerations of their students, a factor that has been identified as being the most significant in the development of creative abilities, according to students.

- By enabling students to manipulate concepts directly, and explore the results of such manipulation, courseware reduces the time taken to understand difficult concepts (Woodhouse and McDougall, 1986).
- Tutorial and drill-and-practice modes seem to be more effective for low-ability students (Gershman and Sakamoto, 1981). Also, courseware is helpful to students reviewing materials with which they had prior familiarity (Dence, 1980).

The major limitations to the widespread use of multimedia courseware and authoring systems in the educational systems are:

- Lack of knowledge among the educators as how to effectively use the computer in an educational setting. Hence, the authoring tool tends to be misused or not used at all. And, in addition, the concern of teachers that authoring system is too difficult for them to learn to use that it is destined to replace them in their jobs. One recommendation to this problem would be to point out to educators that the authoring system and courseware are capable of taking over the routine, information provision and tutorial tasks, and the clerical tasks while freeing the teachers to facilitate learning through one-to-one and small-group interactions (Clement, 1981).
- The lack of sufficient quantity of high-quality courseware is a major disadvantage, which is closely related to the use of an inefficient authoring tool, which lacks in certain high-quality criteria. The problem of portability of software and courseware which restricts the market, the copyright problem, the tremendous amount of time required to develop materials, and the need for experienced and qualified educational and computer professionals are factors contributing to seriousness of this problem.
- Problems associated with the amount of time needed to develop materials and the difficulty of finding qualified and experienced instructional designers and computer programmers still exist.

Special Features of the Authoring System

Based on analysis done by writer, a few features have been identified for implementation of authoring system.

Visual and Object-Oriented Authoring Environment

The primary goal of the authoring system is to create a user-friendly interface that projects usability, simplicity and ease-of-use. The interface selected uses the page/book metaphor. The page/book metaphor is implemented in the authoring module and presentation module, where a sequence of inter-related pages form a book. The interface is designed to give an ease-to-use look to the user, by using the “point-and-click” technique.

- *Media Box and Media Editors*

  It has all the main media types in the form of button icons. The user simply has to select the button to load the media object in concern as visible on the template and use the separate media editors to import the media files. Text tips for each functional button indicates the role attached to it.

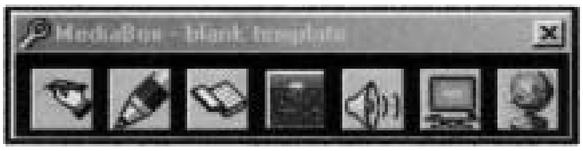


Fig. 1. A sample of media box.

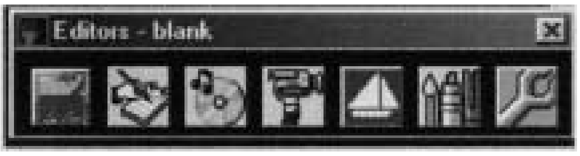


Fig. 2. A sample of multimedia editors.

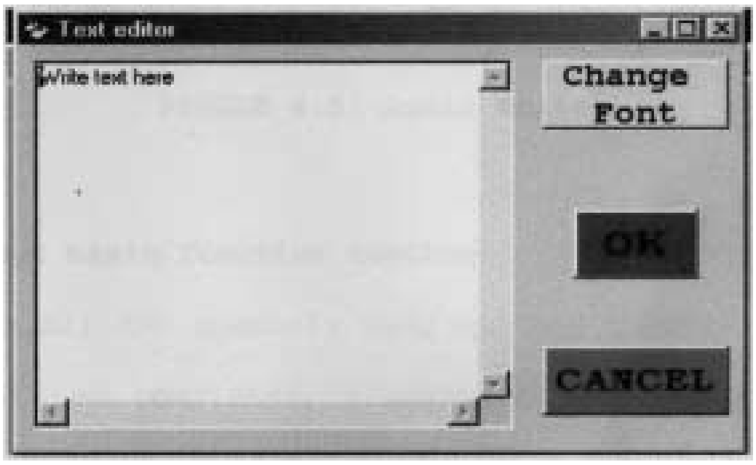


Fig. 3. Text editor.

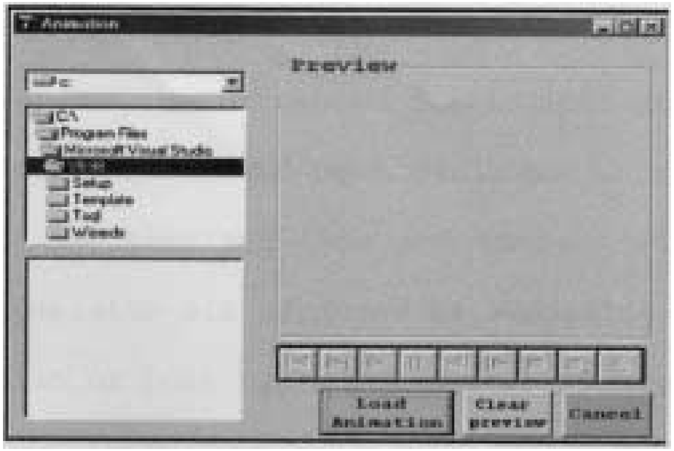


Fig. 4. Movie/animation editor.

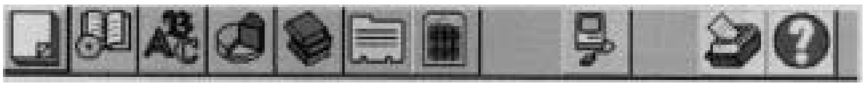


Fig. 5. Toolbar.

- *Short-cut Basic Function Toolbar*
  The toolbar has all the commonly used buttons to allow direct manipulation of new template, arrangement of templates for presentation and the exam bank.
- *Page Arranger*
  This is the dialog box that arranges the logical sequence of the pages or templates. It has been designed to allow text entry for stating the template type and number, whereby the sequence of the templates are arranged as numbering from 1 to 50. Each application or book has a maximum of 50 pages.

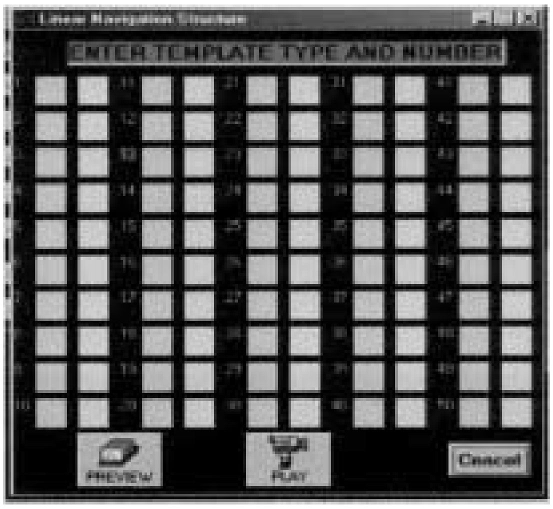
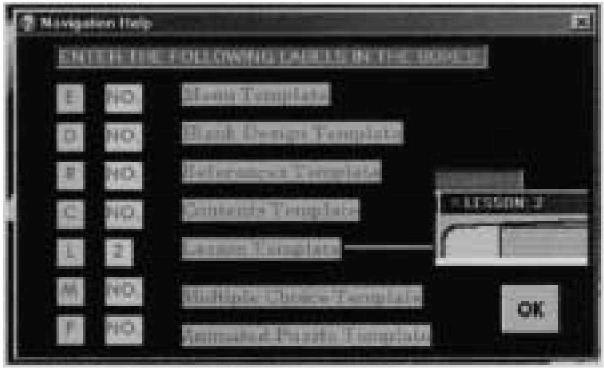


Fig. 6. Page arranger.



E NO Main Template
D NO Blank Design Template
R NO References Template
C NO Contents Template
L 2 Lesson Template
M NO Multiple Choice Template
P NO Animated Puzzle Template

Fig. 7. Template naming convention.

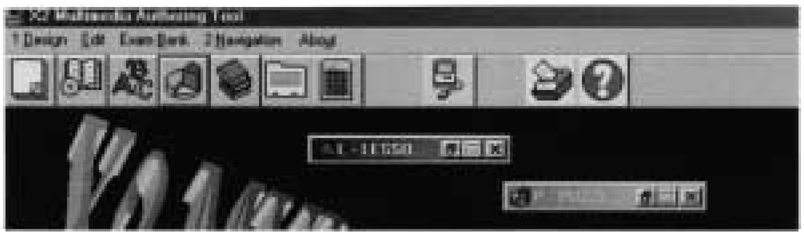


Fig. 8. Authoring module.

- *Design Window*
  This is the window that is also referred as the main module of the system. Multiple pages can be created, to form a book. The design process uses and manipulates the templates akin to manipulation Microsoft Word documents. This feature was introduced to provide the users with some similarity of popular Windows based application. Notably the minimize, maximize and close functions are appended to each template. Majority of users are comfortable with the windowing concept, whereby the child forms belong to a single parent form. This design is yet another familiar feature incorporated in the tool.

Reusability/Object-Oriented Icons

Reuse is the process of integrating existing components into a new application or system, rather than rebuilding everything from scratch. Such a definition refers to the reuse of text fragments, sounds, graphics not specifically produced for the system under development. Retrieval of the object can be obtained by designing and developing a pool of material, which is made available to the application/project developer. Reusability is pursued in the hope that the availability and modifiability of large pools of material can reduce development time and costs. The implementation of reusability of objects in the authoring system is through exam banks which are capable of storing a multiple choice questions and answers.

Multimedia and Graphics

The main purpose of an authoring system is to bring together and link various types of multi-media elements. Due to the variety of media types (text, audio, graphic, images, animation, video), arises a problem of different forms of files for each media types. Usually the more popular and well-known formats will gain wider acceptance among the users than the rest. The issue is the inability of authoring systems to support all the different formats. For example,

- images (bmp, jpg, pcx, tif, tga, gif, dib, rle);
- sound (wav, midi, rmi);
- animation (avi, mpg, mov).

One method of overcoming this problem is by developing an authoring system, which supports all the format types. This will prove to be a huge task and impractical solution, as new formats are being developed all the time. Another common problem is the large size of these files, which requires compression to minimize the size to a reasonable size for storage and transmission. A separate external compression and decompression utility is needed to compress the files during storage or transmission and decompressed for editing purposes.

Reusable Templates

Standard pre-defined templates that are suited for a particular task are designed to assist the user to rapidly design an electronic book or teaching presentation. This tool is equipped with:

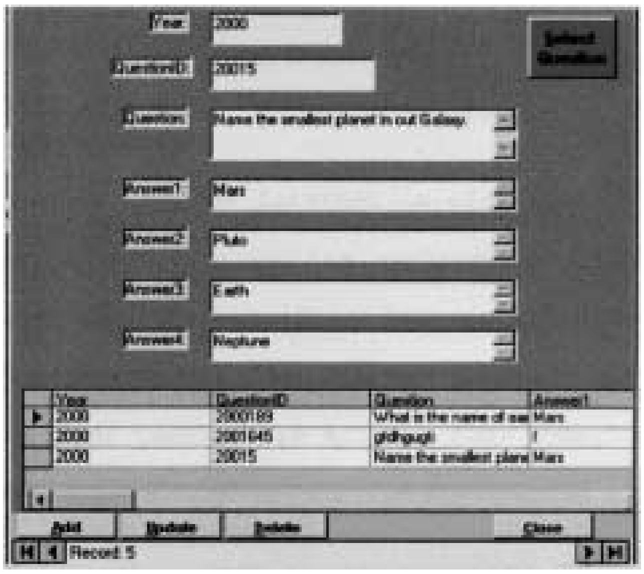
- blank template,
- lesson template,
- multiple choice template,
- multiple choice cum animation puzzle template,
- menu template,
- contents template,
- reference template.

Exam Databases

Database that hold questions, e.g., past year topics, provide the user with an option to randomly select questions from any topic. The database is able to store, retrieve, update, sort and delete records.

Separation of Interface Design and Content Design

Emphasis is on removing the difficulty is using the tool because the system interface has been separated from the design syllabus and lessons content. Content delivery



Year	2000
QuestionID	20015
Question	Name the smallest planet in out Galaxy
Answer1	Mars
Answer2	Pluto
Answer3	Earth
Answer4	Neptune

Year	QuestionID	Question	Answer1
2000	2000189	What is the name of our	Mars
2000	2001645	gfdhgugt	f
2000	20015	Name the smallest plane	Mars

Fig. 9. Input form for science database.

(interface design) has been limited by providing design templates. More emphasis has been placed on the lesson planning and curriculum activity, which in return provides and ordering of the content. Thus, specific authoring tools prove easier and efficient to use than general authoring tools. The range of users can determine certain pages that are information-based (lesson) and then add some quizzes to test the user's knowledge of that content.

Internet Access

In line with the use of the Internet in classrooms and as an online library, the prototype has equipped with hyperlinks that can access the Internet. This feature is noticeable in most of the templates such as lesson, references, and the blank template.

Button-based Interactivity

Both modules, namely the design and presentation uses buttons to:

- select a function or template;
- select and load multimedia files;
- navigate from page to page;
- select an interactive option on the template.

This concept is introduced for the very same reasons why people have the tendency to push a button, press a button or click a button, when they see a button. This behavioral aspect relates to having the end-users naturally link the button commands to being able to do something or activate an event.

Question and Answer Correction and Timer

The templates which employ question and answer sessions are equipped with a timer and an indicator to state the correctness of the answers given by the students or participants during the exercise sessions. The timer will display the amount of time (in seconds) given for a particular question, whereby it closes the question page when the period is up.

Conclusion

The effectiveness of the developed multimedia-authoring tool in producing lesson materials and related tutorials to supplement the classroom instruction is considered an important tool to assist teacher. The features incorporated into the tool support this fact. The authoring tool focuses on its usability, multiple teaching strategies, courseware reuse, and the concept of specialized authoring. The usability factor provides a strategy for target users to utilize the tool without much expertise or training since programming and scripts are not a part of the tool. The tool is designed to be a non-technical and uses a familiar interface that increases usability. Multiple strategies present the domain material in various forms of media and this improves understanding of the material in concern. Conclusively, the tool provides assistance to teachers and students in putting together simple courseware or presentations. This is projected with the use of simple authoring environment, commands and a lower learning curve.

References

Arch, C.L. (1994). *Authoring Interactive Multimedia*. AP Professional.

Chamber, J.A., and Sprecher (1980). *Computer Assisted Instruction*: *Current Trends and Critical Issues.* Communications of the ACM.

Clement, F.Y. (1981). Effective considerations in computer-based education. *Educational Technology*, **21**(4).

Dence, M. (1980). Toward defining a role for CAI: a review. *Educational Technology*, **20**(11).

Gagne, R.M. (1977). *The Conditions of Learning.* New York.

Gershman, J., and E. Sakamoto (1981). Computer-assisted remediation and evaluation: a CAI project for ontario secondary school. *Educational Technology*, **21**(3).

Jona, M. (1991). *Representing and Re-Using General Teaching Strategies: a Knowledge-Rich Approach to Building Authoring Tools for Tutoring Systems*. `http://www.pitt.edu/...al/aied/jona.html`

Skinner, B.F. (1968). *The Technology of Teaching.* New York, Appleton-Century-Crofts.

Vassileva, J. (1992). *Dynamic Courseware Generation: At the Cross Point of CAL, ITS, and Authoring.* `http://www.pitt.edu/...al/aied/vassilev.html`

Woodhouse, D., and A. McDougall (1986). *Computers*: *Promise and Challenge in Education.* Blackwell Scientific Publications.

Young, J. (1984). The case for using authoring systems to develop courseware. *Educational Technology*, **24**(10).

M.N. Ayub is a lecturer at Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. He received his BSc (Hons.) in computer science from University of Malaya in 2000. He obtained MSc in interactive multimedia from Heriot-Watt University, Edinburgh, Scotland. His field of interest includes multimedia support learning, semiotic study for interface and multimedia technology. He is a member of HERDSA (Higher Education Research and Development Society of Australiasia Inc) since 2004.

S.T. Venugopal is a post-graduate student at Faculty of Computer Science and Information Technology, University of Malaya, Malaysia.

N.F.M. Nor is a lecturer at Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. She received her BSc (Hons.) in computer science from University of Malaya in 1999. She obtained MSc in interactive multimedia systems from John Moores University, United Kingdom. Her field of interest includes 3D information visualization, virtual reality, collaborative virtual environment and multimedia databases.

Edukacinės medžiagos sklaidai pritaikytos multimedijos autorinės priemonės kūrimas

Mohamad Nizam AYUB, Santhimathy T. VENUGOPAL, Nurul Fazmidar Mohd NOR

Multimedijos technologijų ateitis priklausys nuo to, kaip intensyviai ir efektyviai bus naudojamos šios technologijos. Švietimas yra viena iš konkrečių sričių, parodžiusių didžiules galimybes panaudojant multimediją. Šis straipsnis supažindina su multimedijos priemonėmis, kurios gali būti puikiai panaudojamos kuriant ar sudarant edukacinę mokymo ir mokymosi medžiagą – vadinamąją mokymo turinio kompiuterinę įrangą (anglų k. *courseware*). Sukūrus šią multimedijos autorinę priemonę, bus sudarytos sąlygos, leisiančios nustatyti multimedijos technologijų branduolį, kuris galėtų papildyti turimus edukacinius išteklius.