Development of a hydrology multimedia courseware

F. Delombaerde, C.A. Madramootoo and B. Mehdi

Brace Centre for Water Resources Management, Macdonald Campus of McGill University, Ste. Anne de Bellevue, Quebec, Canada H9X 3V9

Delombaerde, F., Madramootoo, C.A. and Mehdi, B. 2001. **Development of a hydrology multimedia courseware**. Canadian Biosystems Engineering/Le génie des biosystèmes au Canada **43**:7.23-7.30. A computer assisted learning approach was developed to enhance course material for an undergraduate hydrology and water management course, through the use of multimedia courseware. The courseware was installed under 32bit versions of the Windows' operating system. CD-ROM's were developed as the courseware distribution media. Course material was placed in a Window's graphic user interface using Microsoft Visual Basic versions 4-5. Multimedia files, including images, sound, and movies were added to enhance visualization. Students were able to navigate through the courseware in a non-linear fashion akin to multimedia hyperlink technology. The courseware contained all of the conventional course material, in text format, with multimedia additions. Simulation and prediction tools were added to aid students in problem visualization and problem solving. To modify the course contents, the instructor made changes directly to the CD and re-issued an updated CD to the students. **Keywords**: courseware, computer assisted learning, teaching hydrological simulations.

Une approche d'enseignement multimédia assistée par ordinateur servit à réhausser le matériel d'un cours en hydrologie et gestion des eaux de premier cycle. Le logiciel fut installé sur un système d'exploitation Windows en version 32 bit. Le CD-ROM fut le média de distribution pour ce logiciel éducatif. Le matériel du cours a été intégré dans une interface graphique, utilisant Microsoft Virtual Basic, versions 4 et 5. Les fichiers multimédia, incluant des images, sons et films furent ensuite ajoutés pour aider à la visualisation. Les étudiants purent ainsi naviguer à travers le logiciel de cours d'une façon non linéaire, semblable à la technologie hypertexte. Le logiciel de cours contient tout le matériel du cours conventionnel sous forme de texte avec ajouts de multimédia. Des outils informatisés de simulation et de prévision hydrologique servant à visualiser et résoudre des problèmes hydrologiques, furent également ajoutés. Afin de modifier le contenu du cours l'enseignant modifia directement le CD et rédistribua la mise à jour du CD aux étudiants. **Mots clefs**: didacticiel, enseignement assistée par ordinateur, enseignement de simulations hydrologiques.

INTRODUCTION

Computer Assisted Learning does not replace the instructor, but adds a powerful multimedia instructional tool to the curriculum (Delombaerde and Madramootoo 1997). Hydrology and water management instruction is ideal for multimedia instruction since many principles can be shown through interactive demonstration, hence, an undergraduate hydrology course was chosen for courseware development. Computer assisted learning is already taking place in undergraduate medical education (Greenhalgh 2001), geoscience education (Novak 1999), and engineering (Koronios 1999). The courseware developed was intended as an educational supplement and for viewing as a multimedia textbook with which students can interact. Material is presented in a subject-grouped form, as it is taught within the conventional course.

Computer-Aided Learning (Trevitt 1995) or Computer-Enriched Learning (CEL) (Niemiec and Walberg 1992) can be broken down into several forms: tutorial, drill and practice, integrated performance support system, simulations, interactive exploration, data bank, attention grabber, expert systems and AI, computer adaptive testing, computer managed learning, and an interactive hypertext book (Trevitt 1995). These forms are not mutually exclusive and can be contained within each other.

Multimedia courseware is being used for several purposes: as a teaching assistant or Computer-Aided Learning (CAL) (Trevitt 1995), for in lab use and in an any time, any place (ATAP) format (Hardaway and Will 1997). The ATAP principal is a concept closely linked with CD-ROM type multimedia (Haradaway and Will 1997). With the increased growth of home owned multimedia compliant PC's over the past several years (Reinhart 1995), students can view the course material away from an educational setting.

The in-lab use of multimedia courseware is the review of courseware within an appropriately equipped educational computer lab. The delivery of courseware can stem from either student owned CD-ROMs or network file servers (Hardaway and Will 1997).

Multimedia software, whether distributed via CD-ROM or the internet, provides instructors with the possibility of enhancing (or replacing) their traditional education methods through multimedia demonstration and simulation. The non-linear nature of multimedia instruction can have several advantages compared with traditional teaching techniques. Among the most obvious are the students' freedom to choose the lecture topic of most appeal, which may bolster student interest in course material. Students may also browse course material in its entirety and be aware of upcoming subjects. Through proper multimedia instruction, the student can call on the multimedia educator to repeat lessons and demonstrations as needed.

The demonstration capabilities of multimedia include audio and video playback, as well as image viewing. Streaming video playback, in the form of AVI's (Audio Video Interleaf) allows students to view fully narrated digitized movies within the courseware program, which replaces material otherwise impractical or too expensive to show within a regular classroom or laboratory setting. Three-dimensional animations and stills

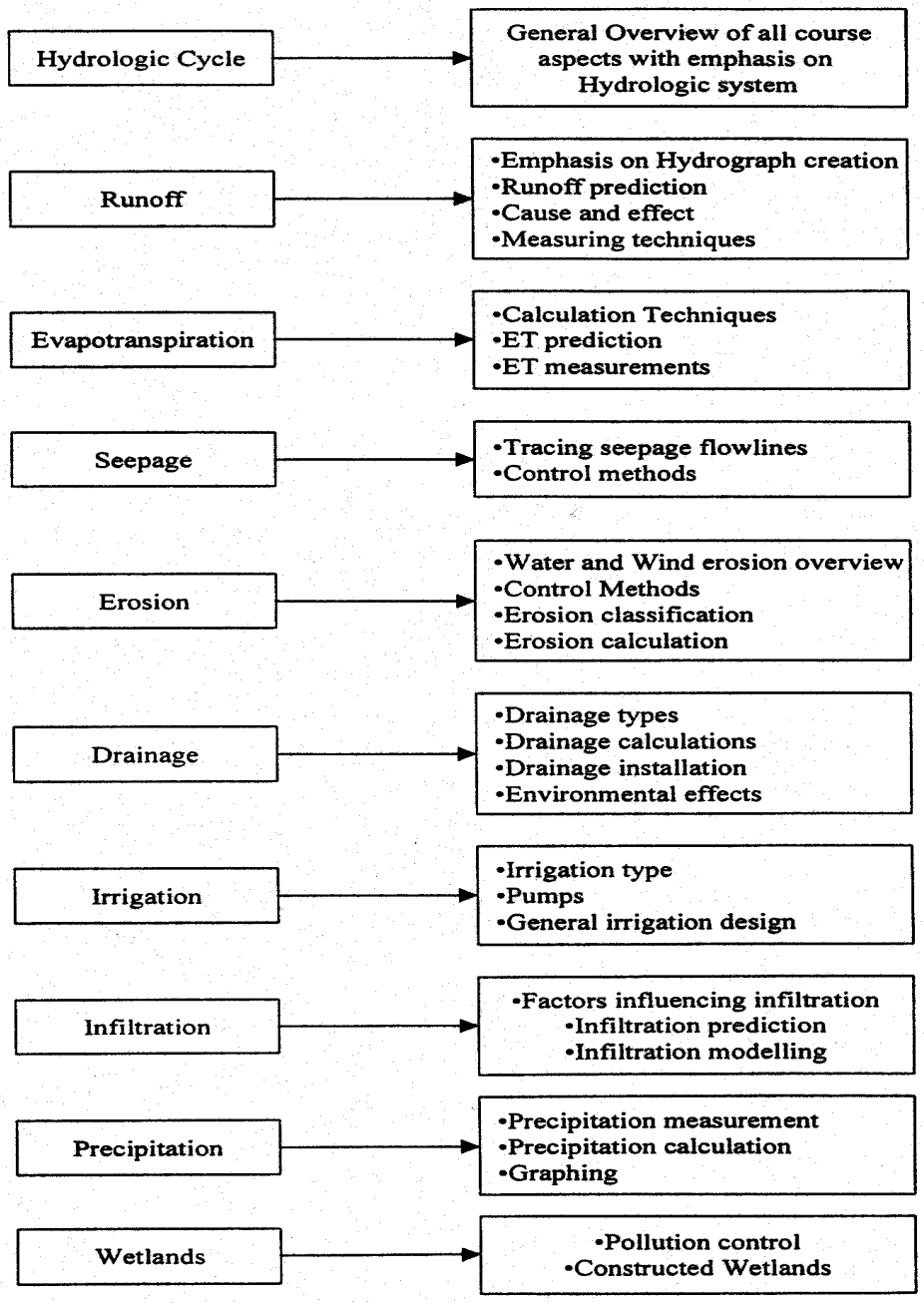

Fig. 1. Courseware material outline.

are suited for explaining processes, which may be too complex to capture using traditional video or camera equipment. These animations can realistically represent events or equipment functioning.

A distinct advantage of emerging multimedia technology is the level of integration of media types within one common interface. Multimedia can be seen as a reasonable substitute for certain real life experiences, which are associated with lab type work (Jensen and Hino 1995). The large capacity (650 megabytes) of the CD-ROM is ideal for holding large files associated with AVI's. Authoring and rendering time, however, are substantial and may account for many hours.

The instructional software was developed for the following reasons:

1. to include text, slides, videos, and photoclips (3D animation) of particular hydrology and drainage projects. A lot of instructional material (text and audio-visuals) had been accumulated over the past 15 years, which would contribute to the richness of the course when incorporated into the courseware. As well, one aim was to be able to download material from the internet and incorporate it into the courseware.
2. to include computer simulation models of infiltration, evapotranspiration, runoff, etc. into the courseware. These were hydrology models which were developed, tested, and used in previous lectures. The aim was to incorporate this accumulated knowledge and tools directly into the courseware in an integrated manner.
3. to maintain the logic and sequence of the existing course structure.

A particular benefit associated with this courseware is the ease with which the course instructor can manipulate multimedia course text. A person with a graduate (MSc.) degree in hydrology/drainage, and who has experience in multimedia presentations and Visual Basic programming can develop a course in this manner. The internet was used to obtain notes, diagrams, information, and data from various sites. This information was then incorporated into the course material, as shown in Figure 1. One new image can also be added per tutorial section even if no images were present before.

Several software packages were used in the courseware authoring process (programming languages, word processors, Internet tools, 3D authoring tools, and multimedia tools). This software was then used to implement the courseware material shown in Figure 1.

The goal of the project was to develop a user-friendly multimedia software to enhance instruction in undergraduate hydrology/drainage courses. The software did not replace the regular course, but was intended for use in conjunction with the regular lectures. It supplements the lectures so that students can learn from difficult concepts on their own, at their own speed.

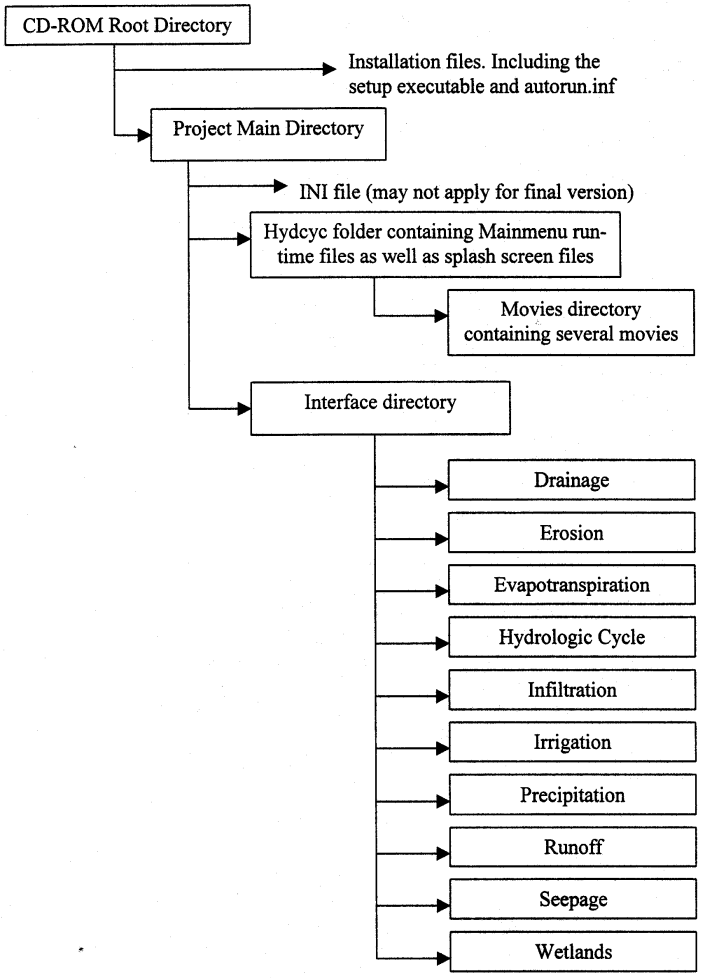

Fig. 2. Program structure.

These concepts include computer simulations of infiltration, evapotranspiration, and runoff. Students were also able to see animation of activities such as drain pipe installation. The courseware was distributed on a multimedia CD-ROM. An advantage of this approach was the ability to update the course material periodically.

COURSEWARE DEVELOPMENT

Courseware navigation

The hydrology courseware was broken down into separate forms, each containing a different subject matter or module. The program follows the hierarchical scheme, starting with a splash screen and jumping to the Main Menu. The user is able to access the program submenus, which allows him/her to select detailed subject matter through tutorials, as well as models and simulations. From the tutorial screen, the user can go back to the sub-menu or to the main menu. However, users can only go back to the sub-menus when engaged in a simulation or module.

Users can navigate the courseware by clicking on hyperlinked words, as well as images and treeviews. Standard Windows-type buttons are also used for navigation, as well as for data entry. Treeviews are used primarily to allow users to shortcut to specific topics or models without having to navigate through the menus.

Courseware structure

Since the courseware CD-ROM contains information executed by the courseware, the directory structure and file locations were made to aid in the development process and not for user perusing. Run-time file mapping consequently matched the directory structure.

During the development, un-compiled forms (which would later make up one single executable file) were contained within the directory from which it obtained the greatest amount of run-time files. This facilitated dealing with complex program directory structure since forms were located in the same directories as their subject matter. During courseware packaging, these forms were removed since they were already contained within the compiled executable file. Removing un-compiled forms also makes obtaining a program's source code much more difficult and requires that the executable file be de-compiled.

Installation and setup files are located in the root directory of the CD-ROM primarily for easy user access (Figure 2). The Setup.exe file is the only executable file located on CD-ROM. Once executed, the Hydrology.exe (packaged as hydrology.exe in the root directory) file was extracted and copied to the host machine's hard drive. All of the files located on the CD-ROM's root directory were extracted and copied to the host machine's hard drive.

Image files, which are installed to the host machine's hard drive, are regularly displayed within the program. By placing image files on the local hard drive, less loading burden is placed on the slower CD-ROM drive when forms are loaded. It also eliminates the need for the CD-ROM drive to load multiple files when loading a form.

For this β-version of the software, the instructor made modifications directly on the CD and re-issued another (updated) CD to the students. The courseware was run directly from the CD-ROM in conjunction with hyperlinks in the various sections for the internet connection. This meant that a computer had to be connected to the internet, if a student wanted to download material.

Hardware required

Visual Basic Programming as well as 3D animations was carried out on a multimedia (MPC2 compliant) Pentium 200 Mhz MMX based system with 64-megabytes of SDRAM and a 4-megabyte ATI 3D-Expression video card. Video capturing was performed using a Power PC 8500 equipped with video capture board, 128 megabytes of RAM and a 180Mhz processor. Other relevant equipment is; Iomega External Parallel Port Zip drive; Kodak PCD Writer; Hewlett Packard CD-ROM writer; Nikon flatbed scanner; Nikon slide scanner; Kodak DC-50 digital camera; Panasonic VCR, and Sony television for video capture and editing.

The CD-ROM version of the software is produced using a CD-ROM writer, which performs a laser burning process, which mimics actual pits and lands in pressed CD-ROM's (Hyon and Martin 1996). The advantage in this process is its ease of production. However, larger quantity output required more time and costs that are higher than batch pressing.

After compilation and file packaging in Multi-Write Publishing Software version 1, course material was placed within appropriate directories and burned onto a CD-ROM, which typically takes about 20 minutes. The compiled program along with the runtime files, such as text, images, and movies are all contained within a CD-ROM. The user now has access to the Setup program, which installs appropriate executable and library files to the hard drive and registers the program in the Windows 95/NT registry.

In order for the program to function properly, the CD-ROM must be inserted in order for the program to access run-time files.

Software required

Several software packages were used in the courseware authoring process. Each package fits into one of five major categories:

1. Programming languages (Visual Basic version 4-5 (Microsoft Corporation)), Word Processors (Word 97 (Microsoft Corporation); Windows Notepad (Microsoft Corporation)),
2. Internet Tools (War FTP Daemon; O'Reilly Website 1.1; Internet Explorer releases 3-4 (Microsoft Corporation)),
3. ActiveX programming tools (Microsoft Visual Java++, Microsoft Visual C++, Microsoft Visual Basic, BORLAND C++, and SYMANTEC C++ (Microsoft Corporation 1997)). Internet Explorer versions 3 or higher to view the Internet download packaged for the second ActiveX component, only accessible to the course instructor.
4. 3D Authoring Tools (3Dstudio release 4 and Max (Autodesk)),
5. Multimedia Tools (Avid Video Shop 3 (Macintosh platform) (Avid); Sound edit 16 (Macintosh Platform (Macromedia); Assymetrix Digital Video Producer (Assymetrix); DDClip (Video editing); Wave Studio (Creative Labs)),
6. CD-ROM Writing Tools (Multi-Write Publishing Software version 1 (Dataware Technologies)).

Operating systems and hardware platforms Although ActiveX components are slowly infiltrating multiple platforms, the courseware, as well as the Internet download component, will only run on Windows 95/98 and NT 3.5 or higher. Consequently, available hardware platforms are limited to those running 32 bit versions of Windows.

Software selection Instructors have the option of selecting pre-made courseware tools, which can quickly produce course material in digital format, or they can develop their material from the ground up. Both methods have their advantages and disadvantages. Pre-made tools save time at the cost of creativity and functionality. Development from the ground up enables for more creativity and functionality, however this takes more time. The ground up approach of courseware authoring was selected for the Hydrology and Drainage class. The course material, student and instructor interface, as well as the multimedia additions needed to be programmed. Even the hyperlink technology, which is available through Internet browsers, needed to be programmed.

Visual Basic was selected as the appropriate programming language with its integration in the Microsoft Fundamental Classes (MFC) (i.e. class constructs for performing basic tasks associated with the Windows' interface such as opening and closing program windows and allocating operating memory). Other reasons for selecting Visual Basic included its tight database integration as well as its ActiveX component authoring and integration abilities. Visual Basic also supports the class constructs associated with object-oriented programming as well as the re-use of ActiveX components, also referred to as objects. On larger scale software development projects, the object-oriented paradigm can serve to organize code more effectively and reduce the overall amount of required code. The software code was authored and packaged within Visual Basic 4-5 Professional Edition. Visual Basic performed several fundamental tasks with respect to the final courseware production.

Other important Internet software are the Web and FTP server software. The O'Rielly Website v.l.l Web server software and the Microsoft Personal Web Server software are used primarily as a server to house the Hydrology and Drainage course information as well as the instructor's courseware upgrade page. Both servers allow administrators to restrict certain pages through password protection, thereby allowing pages containing the instructor's courseware interface to be protected from unwanted student hacking. Teaching assistants can be given access to restricted web pages where they can upgrade course material from any location with an internet connection.

The War FTP Daemon and Microsoft Personal Web Server software were used as FTP servers to house the course text as well as images. The Instructor's ActiveX web page gives him/her access to this FTP site. The FTP site, although password protected, can be accessed by students using the courseware. Required usernames and passwords are contained within the courseware's compiled executable file and are not easily accessible to prying students. The students do, however, have access to, and can modify, the FTP internet address within the courseware, so that the software does not become obsolete if the server IP addresses were to be modified.

To develop the multimedia components of the courseware, several different software packages were used on both the Mac OS and Windows platforms. The Macintosh platform was used primarily in the digitizing of media, whereas the Windows platform was used to edit and package the captured material.

Movie authoring Several steps were taken to capture and digitally package footage from a video camera. After the video was shot, the footage was transferred to a Power PC using Avid Video Shop version 3. This allowed for video quality manipulation and the video size and frame rate. Video was captured in a 320 by 240 pixel frame size by millions of colors at a modest frame rate of 15 fps. Captured movies assumed the QuickTime file format native to the Mac OS.

The movie was made cross-platform compatible in order for it to be used on the Windows PCs. The Mac utility, Movie Cleaner Pro, was used to compress the movies using a variety of compression schemes depending primarily on the type of video which had been shot. This reduced the captured movie file sizes

and facilitated playback on lower end computers which may not be equipped with the proper hardware to decompress video. Proper video compression avoided choppy video playback.

A file conversion tool was then used to convert the movies into the Window's native AVI format. This conversion leaves compression schemes intact, meaning that videos were not required to be re-compressed. The movie files were moved to the Windows platform machines where they were edited using Assymetrix Digital Video Producer and DDClip. The DDClip freeware package was used primarily to paste together separate video clips and add in the voice or sound track, since it allowed immediate playback. At this stage, other video clips, such as 3-dimensional animations or stills were inserted. In cases where sound tracks were used, a voice track was first recorded onto high quality tape and converted to digital W A V format. The sound tracks were then dubbed over the video track. Video clips were arranged to match the recorded voice track. Assymetrix Digital Video Producer was used as a packaging tool for the movies, which had been assembled in DDClip. Movies were designed to play back on a minimum 2X CD-ROM drive or an equivalent of 300Kb/s so as to conform to the MPC2 standard.

3D Animation 3D animations were created using 3D Studio. This authoring tool, running under Windows 95, allows for creation and manipulation of 3-dimensional models over time, creating an animation, which can take the form of an AVI movie clip. Image maps can be pasted to 3D wire frames giving objects a more realistic appearance. More advanced tools, such as changing an object's opacity over time, is also possible allowing for objects to fade in and out of a scene, rendering objects behind them visible.

The majority of objects can be drawn using the primary shapes such as spheres, tubes, cubes, and cones. A Boolean process can be used to add or subtract pieces from the original objects. More complex objects can be extruded directly, or run along a drawn spline path, allowing objects to bend in 3 dimensions. Object’s movements, which are directly related to other objects, such as hinges or a car on wheels, can be assigned a relation to their parent object using a linking tool. This tool sets the hinge points and assigns a certain degree of allowable rotation. Images can then be pasted onto wireframe objects in a process called image mapping. The orientation and suture points of the image on the object can be manipulated as well as tiled.

Once objects have been drawn, a timeline can be used to animate the scene. Objects will start from an initial position at frame 0 and be manipulated over time. A camera must be placed within the scene to create a viewpoint from which to view the animation. Certain animations require that only the camera move so as to provide a "fly-by" perspective of the scene.

Once the animation timeline is complete, it can be rendered to video. Each frame must be rendered separately and written to a single AVI file by 3D Studio. Video is rendered at 320 by 240 pixels at a frame rate of 15 fps. This limits the amount of video quality loss that would occur if the animation were rendered at a higher resolution and then reduced in size using a compression or video editing tool.

Interface

The interface includes every component of a courseware with which a student interacts with either the keyboard or mouse. This includes an appropriate courseware "Start menu" icon. A common and recognizable interface theme must be used in order for students to orient themselves within the structure of the program. This is important at the sub-menu level where students have the opportunity to navigate in any direction.

Four different interfaces were tested at the sub-menu level. The final design allowed students to navigate through either a series of main headings, which often required students to navigate farther in order to find an appropriate subject, or through a treeview, allowing students to navigate directly to a particular subject.

The tutorial sections were developed to accommodate different types of multimedia files. Each tutorial section had a video window contained within it which can access video files at run-time. The video was loaded off the courseware CD-ROM or server only when the user accessed it, either during the tutorial form's load up or by selecting it from the drop down box. The video window was restricted in size to 320 by 240 pixels so as not to overlap with the course text. Color distortion for users displaying 256 colors may cause a problem since certain movies were compressed using a 256 optimized color palette. The video window was packaged with the rest of the program into the executable file. The compression scheme decoders were contained within the video window component.

The tutorial forms also contained three image windows, which come bundled with Visual Basic as intrinsic controls. Like the video window, the image windows access images at run-time to limit the size of the courseware's executable file. Three image windows were inserted into each form to limit the amount of coding required to accommodate images of different sizes and aspects. Using only one image window would have required resetting the window properties each time an image with different properties appeared.

A combo box allows users to select specific subject sub-topics, whereupon the appropriate text and images are displayed. Selecting topics within the combo box does not, however, change the main topic (i.e. runoff will not change to seepage).

A text box displayed context sensitive text. The text is loaded at run-time and displayed as a single paragraph within the text box, which proves to be its greatest limitation. The text box scroll bars will re-size and re-scale automatically at run-time to accommodate texts of different length.

Course software development

The software package for the courseware development was created according to a development structure, or logical development path, similar to program testing.

An Internet ActiveX component was used as an aid in developing the courseware. This was integrated into the courseware itself and was transparent to the user. It served as a means to keep course content up to date. Although all course text is contained within the courseware CD-ROM, hyperlinks were present in the text which allowed the students to access the information on the internet, provided they had an active internet connection.

To convert the conventional courseware into a flow chart format, the main course subject headings were placed within the

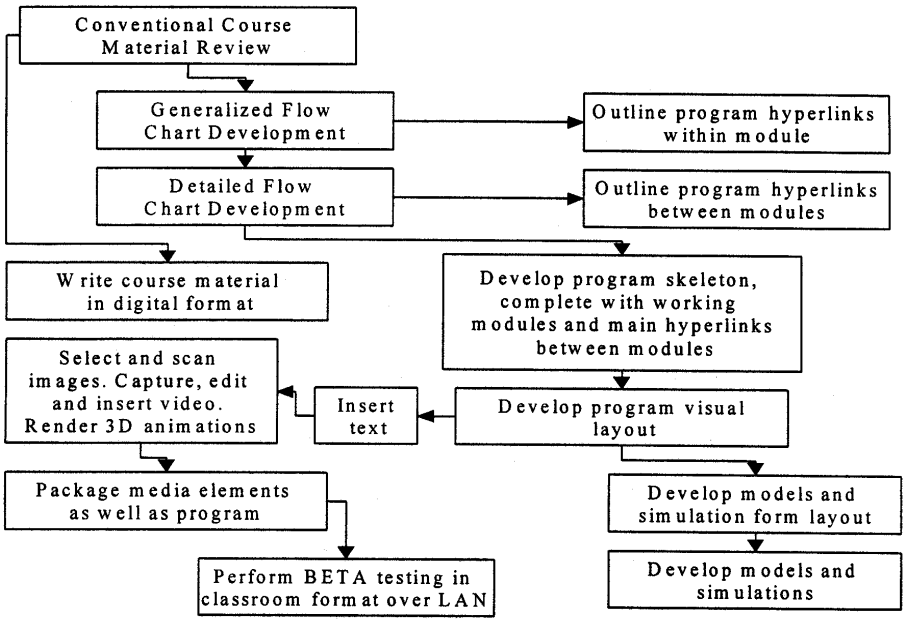

Fig. 3. Development scheme.

flow chart as individual modules (Figure 3) in a hierarchical scheme. Main module hyperlinks were inserted (within the flow charts) to ensure that a user could travel up and down the hierarchy at any location within the program. With the main subject groups and hierarchical hyperlinks, it was possible to proceed with sub-dividing main modules into individual tutorial modules. Course text could be converted into digital format. Each tutorial module was associated with one section of course text. Lateral hyperlinks or links between subject modules (as opposed to hierarchical links) were inserted into the more detailed flow chart scheme, linking different subject tutorial sections.

The next step in the development was the implementation of a working program skeleton, to follow the outline developed during the first stage of the flow charting process. The program consisted of a main menu with links to all of the main course subjects' sub-menus. These sub-menus were linked to the appropriate tutorial sections. At this stage of development, no text, images, or lateral hyperlinks were included.

A visual layout of each form was developed with a standardized "motif", to orient users within the program's hierarchy. Each sub-menu had distinctive characteristics allowing users to discern between different course subjects with visual clues, such as representative images and menu and toolbar titles. The tutorial sections, however, were quite similar in layout and were only discernable through background image and toolbar titles. The written text was inserted into appropriate tutorial sections and lateral links between modules were implemented.

Although models were integrated into the courseware package, they could also be used as stand-alone packages, which could be accessed from the Start Menu within the courseware program group. Once in the courseware, they could be accessed from both the sub-menus and, in some cases, from within the tutorial sections. The tutorial form loading process is shown in Figure 4.

Image selection Images from a bank of slides related to hydrologic studies were selected and scanned in a high quality Nikon slide scanner and saved as BMPs or JPGs in high colour format (16-bit color). Since image windows within the tutorial modules are no larger than 400 pixels high by 400 pixels wide, images sizes were reduced. The image containers within the Visual Basic interface would automatically resize, keeping the upper left-hand corner of the container anchored, to fit images with different aspect rations.

APPLICATION TESTING

Program size

Since the delivery tool for the courseware was a CD-ROM, the overall program size needed to be contained within an abundant 650 megabytes, which proved to be feasible. The components, which require disk installation need to be limited in size. Furthermore, the program executable file must be restricted in size so as to assure rapid execution.

The total installation, including system and library files, was approximately 7 megabytes, 4 of which represented the executable file. The executable file was installed in a separate directory specified during installation. The library and system files were automatically installed to the Windows system directory where they were registered. An installation log was created during setup which allowed users to uninstall the program using the Windows' uninstall utility (accessible through the Control Panel). The uninstall utility removed the program executable directory and all files contained within it. It also "un-registered" the program from the system, removing the "Start Menu" icon. The uninstall process, however, did not remove the settings which were created within the Windows' registry.

The remainder of the program, constituted largely of run-time files, remained on the CD-ROM. The program, at startup, located the CD-ROM and read the appropriate information from predetermined directories.

The multimedia support offered by Microsoft Visual Basic (Microsoft Corporation) versions 4-5 and C++ as well the programming languages' flexibility and ActiveX (Microsoft Corporation) support, make these packages ideal for authoring complex multimedia courseware products. The interchangeability and reusable nature of ActiveX controls allow for easy manipulation of components written in C++ or to Visual Basic and to the Internet. Microsoft has also distributed reusable Internet and Network ActiveX components through the release of their Internet Explorer package.

A host of basic components are packaged with Visual Basic to give developers access to a variety of pre-made components which facilitate simple processes, such as creating and

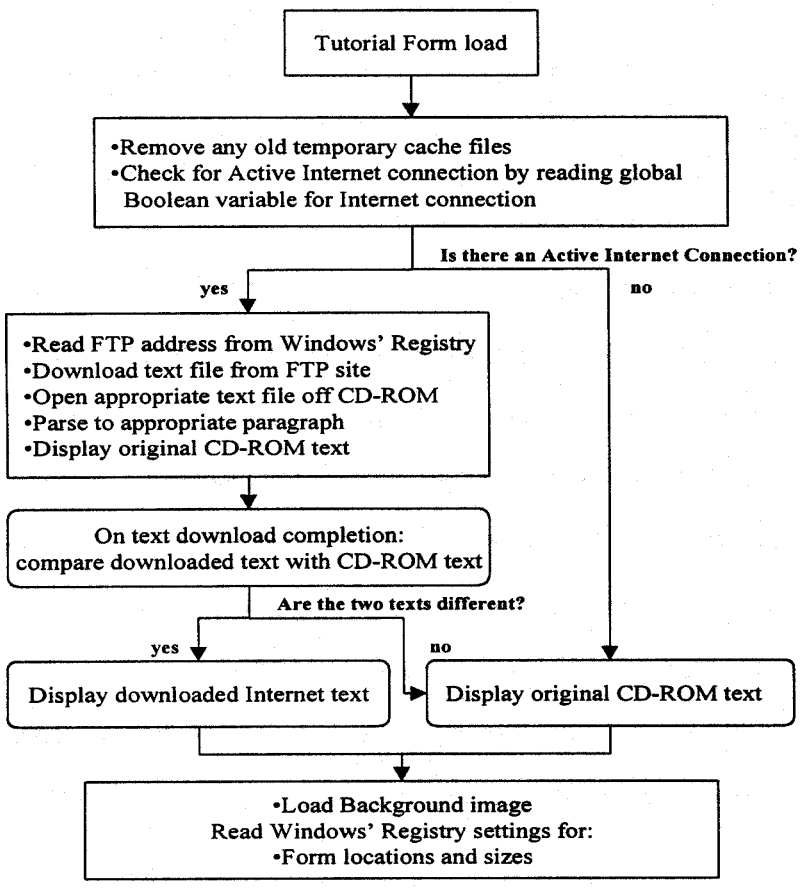

Fig. 4. Tutorial Form Load process flow chart.

manipulating text boxes and image windows. This allows programrs to spend less time with the inherent overhead of a visual interface and more time on program development. Downloadable and third-party ActiveX components further simplify complex programming tasks by presenting developers with fully functional and integrated software codes.

Setup program The courseware setup utility serves several essential tasks. Primarily it serves to extract installable courseware files and copy and register them to the host hard drive and OS. This process is mostly automated and comes standard with Visual Basic. The setup utility, which comes bundled with Visual Basic, is essentially a stand-alone executable file, which performs file extraction and installation and ActiveX component registering. It also creates a program group in the Windows' Start group. However, non-built in functions must be added. In the case of the Hydrology courseware, several functions were added in order to make the setup program slightly more versatile from a developer's point of view. When installing files which will be accessed at run-time to the hard drive (i.e. *.ini files) it is important that the program know where the run-time path is found. This, however, is not a straightforward process since the user can change the installation directory during the program setup. The setup program must leave some trace of evidence that the courseware executable file can find, which tells it where to find certain run-time files. Writing the directory information to the Windows registry can accomplish this. During courseware setup, the user must select an appropriate directory and drive for which to install the courseware files. At this time the chosen directory is recorded within the registry. During program execution, the registry information is extracted by the courseware executable and stored within system memory as a string variable.

Program performance

Several different executable file configurations were tried in order to maximize the startup and running speed of the courseware. Emphasis was placed on the rapid display of the program's splash screen, which indicated to the user that the program was loading. In an effort to limit the total executable file size, most images were not bundled with the executable file, except for the splash screen which was bundled to speed up the display. The main menu form was loaded from a timer running in the splash screen so that execution speed was identical for all machines, regardless of performance. In an effort to reduce run-time lag, most image and movie files were compressed.

Performance changes were difficult to measure when running off a CD-ROM given to the greater bandwidth of CD-ROM's. However, a host of other factors came into play, including RAM, CD-ROM drive cache, and the "CD-ROM read-ahead optimization" settings within Windows 95/98iNT, which served as a secondary cache for the drive. Regardless of these factors, the program's overall performance increased with smaller run-time files.

Instructor's courseware web page

Execution speed was highly variable, depending upon both the machine speed and the ActiveX container. Newer versions of Internet Explorer, which featured OS integration, consequently ran the ActiveX component significantly faster than older, non-integrated versions, although both executed the component relatively quickly.

The ActiveX component had an error catching code inserted into it. The error handling consisted primarily of returning error codes when courseware directories could not be located. Specific codes returned pop-up message boxes, which prompted the instructor to enter the courseware CD-ROM or to select a different drive. No error handling however was added for the FTP component, which would not inform the user of a bad Internet address or a silent server. This was intentional to avoid adding extra error handling codes. No uploaded text indicating to the user that an error had occurred was displayed on the text window when a server error was encountered.

If an offline server became active after an error occurred, the instructor was required to change course subjects using the dropdown combo box to re-download text.

An "Add Image" option was available to the instructor in the ActiveX component contained within the web page. By pressing the "Add Image" button, a "Save As" pop-up directory window appeared. The file types to be saved were restricted to *.bmp, *.jpg or *.gif so as not to induce run-time errors when users uploaded images. Once the instructor had selected the appropriate image, the file was copied, renamed and placed on the FTP server. The name given to the file corresponded to the location within the text file with which it was associated.

The instructor's ActiveX web interface was approximately 55K (took approximately 20 seconds to download using a 28.8Kbps modem) and resided on the user's hard drive in the form of an *.ocx file after being downloaded once. The ActiveX component's size was pre-determined and could not be scaled within the web browser. Instead, the web browser automatically inserted vertical and horizontal scroll bars as required.

CONCLUSION

The final courseware proved to be user friendly, covered a detailed amount of material, and presented the material in a clear manner to the user. As well, the material was of educational relevance and could be kept up to date in two ways. The hyperlinks within the CD-ROM text allowed for the users to access the latest material on the internet, provided they had an active internet connection, or the instructor could update material on a new CD and issue the latest CD to the users.

Several factors need to be taken into account when considering the overall courseware development, such as outlined below.

The software engineering aspects must include a level of software professionalism (including the coherence with today's professional software development standards); a degree of hidden programming, giving the user easy to use, yet powerful tools and information; as well as an ease of program installation and removal.

Courseware requiring more time to complete often contains a higher degree of these software characteristics. The output quality of both the final courseware and the multimedia components contained within that courseware dictate the amount of time required to author an entire course. The greater the quality and quantity of multimedia material, the more time must be invested in the development phase.

The degree of hidden programming is also a major factor to take into consideration during development. Although development of the initial interface can progress quite rapidly, adding functionality to the interface can be demanding. In an effort to reduce the overall time required to add functionality, an Object Oriented approach can be helpful in cases where functionality must be added to similar forms. Visual Basic is adept at handling class constructs as well as creating ActiveX components, which is coherent with the Object Oriented paradigm.

REFERENCES

Delombaerde, F. and C. Madramootoo. 1997. Enhanced hydrology instruction using multimedia. In *Special Proceedings of the ASAE (Mini-Conference): Applications of Emerging Technologies in Hydrology*, 95-98. St. Joseph, MI: ASAE.

Greenhalgh, T. 2001. Computer assisted learning in undergraduate medical education. *British Medical Journal* 322(7277):40-44

Hardaway, D. and R.P. Will. 1997. Digital multimedia offers key to educational reform. *Communications of the Association for Computing Machinery* 39(4):43-45.

Hyon J.J. and M.D. Martin. 1996. CD it for yourself. *BYTE* 21(6):105-112.

Jensen, E.C. and J.C. Hino. 1995. Multimedia instruction: Creating new learning environments. *Journal of Forestry* 93(5): 8-14.

Koronios, A. 1999. Integrating instructional design guidelines in courseware engineering. *Journal of Educational Multimedia and Hypermedia* 8(3): 315-330

Microsoft Corporation. 1997. Microsoft ActiveX: Developing Exciting Content and Applications for the Internet and Intranets. Microsoft Developer Network, Library-Visual studio 97. Mk:@ivt:backint/D1/S983.HTM.

Niemiec, R.P. and H.J. Walberg. 1992. The effects of computers on learning. *International Journal of Educational Research* 17:99-107

Novak, G.A. 1999. Virtual courseware for geoscience education: Virtual earthquake and virtual dating. *Computers and Geoscience* 25(4):475-488.

Reinhardt, A. 1995. New ways to learn. *BTYE* 20:50-72.

Trevitt, A.C.F. 1995. Interactive multimedia in university teaching and learning: Some pointers to help promote discussion and design criteria. Presented at Computers in University Biological Education Virtual Conference. Liverpool, U.K. January 30-February 10.