

Use of Multimedia Development Software for Engineering Courseware

Robert Abbanat, Kurt Gramoll, James Craig
Georgia Institute of Technology

Introduction

Over the last several years, the integration of multiple media sources into a single information system has become a rapidly developing phenomenon [1,2]. This growing technology, which many refer to as multimedia, merges graphic, audio and video data into a computer-based information retrieval program. Until recently, it has been difficult for people with non-computer science backgrounds to develop and author sophisticated multimedia programs. However, with the recent introduction of many high level authoring, animation, modeling, and rendering programs for both the Macintosh and Windows-based computers, significant multimedia courseware programs can be developed by professors and students [3-6]. Even with these new media tools, a considerable amount of time is required to produce an interactive multimedia program for engineering education.

The School of Aerospace Engineering at the Georgia Institute of Technology, with support provided by the National Science Foundation, has initiated a multimedia research and development program to develop solid mechanics courseware using various multimedia technologies. The primary goal of the engineering courseware is to illustrate a number of basic concepts in introductory engineering courses using the application of integrated video, speech, and three-dimensional graphics in computer-based, self contained software modules. These modules are developed to serve as a supplement to lectures and textbooks, illustrating topics befitting the audio and visual capabilities of multimedia and not to replace the instructor or textbook. Data visualization [7] is an important factor in the use of multimedia to improve on the standard learning method of lectures and textbooks.

There are many technical issues, practical considerations, and obstacles involved in the authoring of engineering courseware using multimedia. This paper discusses recent findings concerning the use of multimedia for solid mechanics courseware and presents some of the techniques and methodologies that have developed at Georgia Tech during the past two years. Various multimedia tools are reviewed for their effectiveness in the development process. In particular, current authoring, modeling, rendering, and animation software is discussed for its usefulness in

constructing engineering courseware.

Engineering Courseware Topics

Before the actual software used to construct multimedia programs can be discussed, the content of engineering courseware and the courses that can use it must be addressed. The engineering concepts best suited as a topic for an engineering courseware module are those which are highly abstract and difficult to visualize, such as a 3D stress tensor, or are dynamic, such as complex structural dynamic response. Concepts which are static, like a two dimensional bending moment diagram, can still be presented in engineering courseware, but often are not as stimulating. Although a simple 2-D bending moment diagram can very easily be shown in a text book, multimedia methods can link it to a video of a practical situation or illustrate how changes in loading affect the diagram. The presentation of highly abstract concepts which may have no physical form can be facilitated and enhanced using multimedia. Consider, for example, using multimedia to illustrate the behavior of a 2nd order tensor as it is used to represent stress or strain. In this case the multimedia techniques used may be closely linked with scientific visualization methods.

When choosing a topic for a module, the developer must also consider the education level of the students who will be using the courseware. Basic Newtonian mechanics presented in high school physics are simple enough that descriptive animations could easily be created. The same could not be said for the trailing vortex behind a finite airfoil moving through a fluid, which is usually covered in third-year undergraduate aerospace engineering courses. In general, the level of difficulty involved in multimedia development increases as the sophistication of the topic increases which in turn changes the type of software and hardware needed for development.

Engineering Courseware Development Process

The processes that generally occur in the

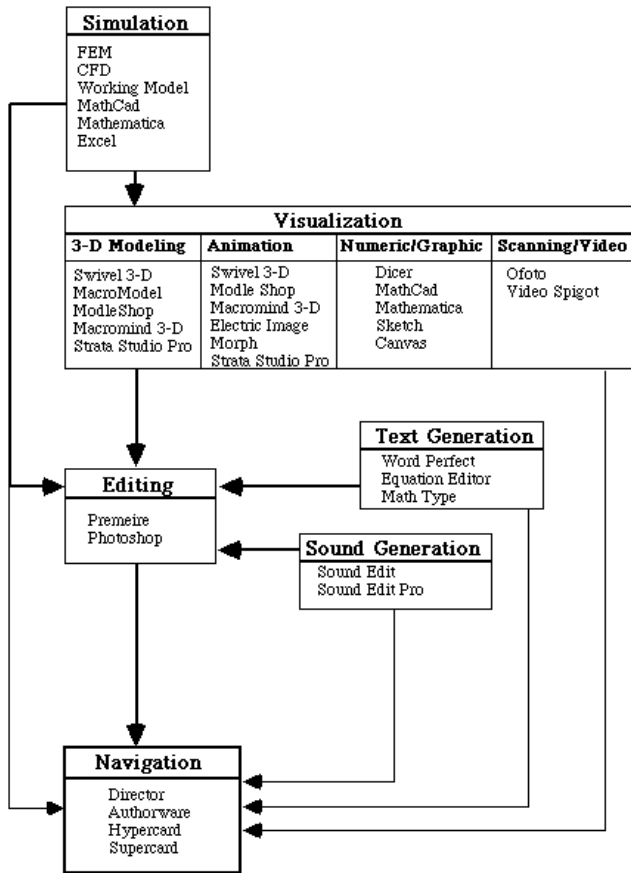


Fig. 1. Flow chart of Engineering Courseware Development Processes.

development of an engineering courseware module are nonlinear, as shown in Fig. 1. Although numerous multimedia software programs are needed, they can be separated into categories based on functions performed, such as simulation, modeling, rendering and authoring. Once an engineering topic has been chosen for implementation, the developer will in many cases, need to first simulate a problem or situation that illustrates the topic, then transform output from the simulation into an appropriate visual form, define animation to describe the system, render the animation, generate audio and text components, define the navigation topology and finally integrate everything using an authoring package. Depending on the nature of the topic, not all of the processes listed above are required for every module. Frequently a software package, such as Working Model or Strata Studio Pro, will have the capability to perform many of the tasks simultaneously. The chart shown in Fig. 1 is not intended to serve as a recipe, but only as general guidelines.

Simulation

In most cases, it is unlikely that a developer will find a working simulation of an engineering problem available on a computer. Thus, for many topics, the first step in the development of a engineering courseware module is the creation of a working simulation of a problem that can be analyzed with some degree of accuracy. This not only

provides the developer with the ability to tailor and modify the model, but it reduces the possibility of copyright infringements.

There are several packages designed for engineering analysis and animation simulation. Working Model, for example, merges advanced motion simulation technologies and an interface with flexible editing abilities to allow a student to study Newtonian mechanics using direct experimentation with real world situations. An infinite number of scenarios can be analyzed, and the output can be in the form of animations, graphs and charts, or numerical data. Software packages from other sources allow for analysis of mechanical systems using the finite elements, CFD (computational fluid dynamics), and numerous other mathematical analysis methods.

A developer might, for example, choose Working Model to create a simulation of a spring-mass-damper system. The mass, spring constant, and damping coefficient could all be specified along with an applied force. Working Model would then generate output which accurately depicted how that system would behave. The output could be in the form of animation showing the motion of the system, charts and graphs showing the forces, moments, accelerations, velocities, and positions of the system, or numerical data describing to the forces, moments, accelerations, velocities, and positions of the system.

Visualization

Effective visualization is the key element in the presentation of an engineering topic using multimedia. In some cases the output of simulation programs is not well suited for visualization and, as such, is only useful to the well trained engineer. This usually occurs when the output is in the form of numerical data that must be transferred to another program for further modeling and animation. In other cases it is better to begin development of a module with a 3-D modeling or animation package, bypassing the creation of a simulation.

3-D Modeling

Three dimensional (3-D) models can be created to provide enhanced visualization of components involved in an engineering problem. This is done by specifying the basic dimensions of the component in 3-D space and then using solid modeling methods to define surfaces and create volumes. Complex models can easily be created by combining several simple models. Some modeling programs, such as Model Shop, Swivel 3D, Macro Model, Sketch, and Strata Studio Pro, allow the user to create unusual models by manipulating the control points of a spline-based curve or surface. In general, multimedia geometric modeling programs are less precise than traditional CAD programs. This is usually not a problem, however, since the models are not used for manufacturing purposes. In some cases, the data from a simulation can be transferred directly to the 3-D modeling program, and in the process save time and effort.

For one courseware module created at Georgia Tech, a student used Macromedia 3-D to create a model of the Space Shuttle. This model was combined with a second model of

the Earth, and used to create a realistic scene depicting the orbit of the Shuttle around the Earth.

Animation

Because engineering is often concerned with dynamic processes, animation is paramount to the effective visualization of an engineering problem. Once the components of a system have been created using 3-D modeling packages, the engineering problem can be brought to life with animation and rendering. Most packages that perform advanced animation also have advanced rendering capabilities. Some animation/ rendering packages do not support 3-D model creation, hence the model must be created elsewhere and imported.

A motion path, which specifies the behavior of the objects of an engineering problem, can be created by specifying the position and orientation of each object along a time line. Objects include lights and cameras as well as the components (3-D models) of the problem. Since it is not practical to specify an object's precise location at every instance in time, the developer can usually specify position and orientation at specific locations, or key frames, placed at various intervals along the time line. The program then interpolates the motion of the object between the key frames. Some programs, such as Gryphon Morph and Macromedia 3-D, allow the user to specify the shape of an object at key frames. The object will change into a different shape, or "morph", as it travels along its motion path.

In some cases the motion path can be specified using data that was created with the simulation. Simulation programs such as Working Model will output the position and orientation of the components of a system in numerical tabular form. This data can be directly imported into some modeling programs to define the motion path. This method produces more accurate and realistic results than specifying key-frames as described above.

For example, a developer might import the numerical data from the spring-mass-damper simulation into an animation package such as Macromedia 3-D or Strata Studio Pro. This data could be used to define the motion path of a three-dimensional mass within the animation package, thus creating a realistic 3-D simulation of the spring-mass-damper system that could be observed from any angle. Alternately, the developer could attempt to imitate the motion of the mass by specifying key frames along the timeline. It is unlikely, however, that the degree of accuracy of the motion displayed by the imported data would be matched.

Rendering

Once the objects have been created and the motion path has been specified, the scene is rendered. Rendering programs will usually generate a series of frames (stored in PIC files) showing the object as it travels along its motion path. The user can specify the surface characteristics of an object, for example, color, texture, pattern, luminescence, transparency, and reflectivity. Most programs will allow the user to specify animation at between 1 and 30 frames per second, as well as the size and color depth of each frame. The output is usually a QuickTime Movie or a PICS file.

Depending on the complexity of the model, the length of the motion path, and the size and complexity of the frames, the required rendering can be extremely costly in terms of time and memory.

Scanning and Video Recording

Several other methods are available for the acquisition of visual information in electronic form. A scanner can be used to convert a 2-D picture, such as a sketch or photograph, into a PICT or bit-map file. Scanners have a varying range of resolution (from 100 dpi to 1200 dpi) and operate much like a fax machine. Software, such as Ofoto, drives the scanner and allows for some basic editing of a scanned image. Video Spigot is a software/hardware package that converts analog information from a camcorder or VCR into digital format. The output of Video Spigot can be a QuickTime movie or PICS files. This provides an alternate method of producing animation, the results of which are photo-realistic.

Different multimedia development packages perform different tasks, but no single package provides all the needed capabilities to create an engineering module. This is especially pertinent when considering all the packages available for visualization. Model Shop and Macro Model are used to create models. Swivel 3-D can create models, specify motion path and perform basic rendering, but it does not allow for the specification of surface characteristics. Electric Image is excellent for specifying motion path and surface characteristics, but the model must be created elsewhere and then imported. Macromedia 3-D can create basic shapes, specify surface characteristics, specify motion, and render, but it is somewhat cumbersome and the lighting features are weak. Strata Studio Pro, with good capabilities in all areas, is perhaps the best all-around package.

Sound and Text Generation

When used properly, an engineering courseware module can be greatly enhanced with the addition of sound and text. Music and voice annotation can make a module entertaining and captivating. SoundEdit is a product from Macromedia that can be used to create audio clips. Sounds can be imported, or created using a microphone. SoundEdit has a variety of filters and effects that can be used to create almost any type of sound.

Text can be used to incorporate some of the theory that is found in textbooks and lecture notes into a module. Text and equations can be created in word processing packages and then imported. Currently, only a fraction of the text found in a textbook is being integrated into engineering courseware modules at Georgia Tech. As stated earlier, the strength of multimedia is its audio and visual capabilities. Further, it is recognized that people tend to dislike reading excessive text on a monitor.

Editing

Once a frame, series of frames, QuickTime movie, or audio clip has been created, there are various packages available for editing and modification. Adobe Photoshop has

tremendous capabilities for editing and modifying a still bit-map image. In addition to basic drawing functions, Photoshop can re-size, transform, filter, clone, convert, or generally manipulate all parts of a still image. Adobe Premiere is available for video editing and the creation of QuickTime movies. Premiere allows for the merger of several video and audio clips or tracks. Transitions can be applied between tracks, images and titles can be superimposed, areas can be filtered, textures can be applied, films can be rotoscoped, insets can be created, and screens can be split.

Adobe Premiere would be an effective tool for presenting the various stages a rod undergoes (i.e. elasticity, plasticity, strain-hardening, necking) when acted upon by a tensile, axial force. Animation showing all stages through to fracture could be imported. Arrows, text, diagrams, and audio clips could be overlaid and made to fade in and out at appropriate intervals during the animation.

Navigation

Once images, animations, and sounds have been gathered, they are combined in an authoring package that allows for navigation. The authoring package is used to define and shape the actual courseware module. The authoring packages that have received attention include Macromedia Director, Macromedia Authorware, HyperCard and SuperCard. All perform similar functions, but Macromedia Director, though complex, is the most powerful and sophisticated of these.

Within Macromedia Director, text, still images, audio files and QuickTime movies are stored as individual members of the *cast*. All the output of a Director module is viewed on the *stage*. The structure of a Director module is established in the *score*, which is similar in concept to a conductor's score. Members of the cast are placed in the score. Time along the score is measured in *frames*, each frame containing 24 *channels*, so that as many as 24 different cast members can appear in one frame. When the *playback head* of the score reaches a frame containing a particular cast member, that member will appear on the stage.

Director also has its own scripting language called Lingo. Lingo can be used to control the movement of the playback head, or to create procedures and functions for input and output. Lingo has the power to perform calculations, much like Pascal, FORTRAN, or any other high-level procedural programming language. This is the basis for user interactivity. At a particular point in the score, Lingo might, for example, prompt the user for the dimensions of a rod and the forces acting on it, and then generate animation to illustrate the effects of those forces on that rod. In this instance, Director would not actually be generating animation, since it does not have rendering capabilities nor can it create images in real-time. Several different animations depicting a

rod stretching must have been created in another package such as Strata Studio Pro, and placed in the cast. Lingo can calculate how the rod should deform based on the given input, and then display the appropriate previously created animation that most closely matches the situation. Director's lack of ability to create images in real-time is due to hardware speed limitations in the host computers and has proven to be a limitation since it reduces the accuracy of the information displayed in an interactive module.

Structure and Appearance of Engineering Courseware

The previous sections have focused on engineering courseware topics and production and development processes, but there are equally important issues involving the structure, layout, and appearance of engineering courseware. A brief description of some of the techniques and methodologies that have worked well in engineering courseware modules is presented next.

Depth of Courseware Module

Once the subject of the module has been chosen, the general structure of the program must be developed. A "slide-show" type module or presentation program will illustrate a concept with a series of scenes, each containing a still picture, animation, graph or dialogue. The progression of a slide-show module is linear, in that the user starts at the beginning and follows the scenes in a sequential order. On the other hand, an "interactive" module will allow the user to specify or change parameters and then react accordingly. The user might, for example, specify the dimensions of a rod and the forces acting on it, and the module would generate animation to illustrate how that particular rod would deform under the prescribed forces. Interactive modules are less linear in nature, and they encourage exploration and experimentation. Interactive modules are much more difficult to develop than simple slide-show modules, but for engineering, interactivity and the ability to proceed to other topics quickly is paramount. Students in technical disciplines require the ability to interact with the current design or analysis problem and perform what-if evaluations to fully comprehend the problem parameters.

If a module is to be used as a supplement to a particular class, the developer may also wish to consider its correlation with a text book. In this case, the material in the module could be presented to coincide with specific chapters in the book where the pictures and diagrams in the text book could be animated. In addition, narration could be included in a module which closely following the text. In these situations, copyright issues must be considered.

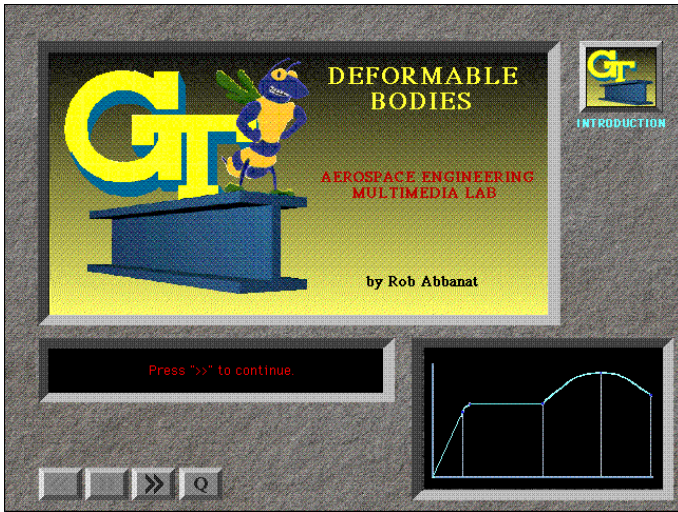


Fig. 2. Stress/Strain Module. Introduction Screen.

Interface of Courseware Modules

Screen layout is an important issues since it is part of the direct interaction process between the user and the module. Entire courses are devoted to human/computer interface, the details of which are beyond the scope of this paper. Only the key methodologies that have received attention at Georgia Tech are discussed here.

The use of panels, such as the ones in Fig. 2 and Fig. 3, is an efficient and logical way to break up the "stage". Panels may contain several windows, each of which is used to display a different type of information. The larger window in Fig. 4 is used to display the main event in a scene, such as the rod stretching under a prescribed force. The smaller window in the lower right of Fig. 4 is used to display information that plays a secondary role in a scene, such as the progress of the rod along the stress/strain curve. The long, narrow window is used to display textual information and instructions. It is recommended that the background be a darker color (black in Fig. 4) and the foreground a lighter color (gray in Fig. 4).

This brings the face of the panel forward, and provides a contrasting background for the animation and text to be clearly visible in the windows. Colors that are too bright in the panel may distract the user's attention from the information being displayed in the windows. Beveled edges are used to give the panels a 3-D appearance. The panels remain the same throughout a module in order to create a sense of continuity and consistency. Thus, in any scene, the user will know where to look to find a particular type of information. This is reinforced by a logo that appears in the same spot in every screen, such as the square in the upper right of Figs. 2 and 4. This continuity can be carried over to other modules covering similar or related topics.

The navigation through a module can be controlled with navigation buttons similar to those shown in the bottom left of Figs. 2, 3 or 4. These buttons operate in a similar fashion to those found on a VCR, with standard icons indicating forward, reverse, and stop. A QUIT button should be incorporated to allow easy exit from the program. If the module contains a main-menu, a button should be incorporated that will return the user to the main menu from any part of the module.

Hardware for Engineering Courseware

Understanding the technical aspects associated with the hardware is an essential factor for efficient production of engineering courseware. With regard to graphic capabilities, Macintosh machines have consistently had an edge over their PC counterparts. As a result, virtually all the software previously mentioned runs exclusively on the Macs, making them the machines of choice. On the other hand, most good engineering analysis software is available only on the PC. Some packages such as Macromedia Director, however, are scheduled to appear with versions for the PC. Until the graphic capabilities of the PC improve and all multimedia programs arrive with PC versions, the Mac appears to be the primary machine for multimedia productions.

Two types of machines must be considered when

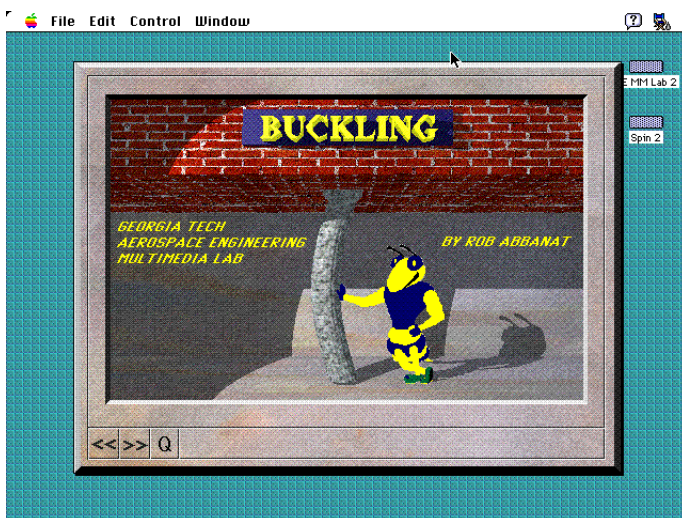


Fig. 3. Buckling Module. Introduction Screen.

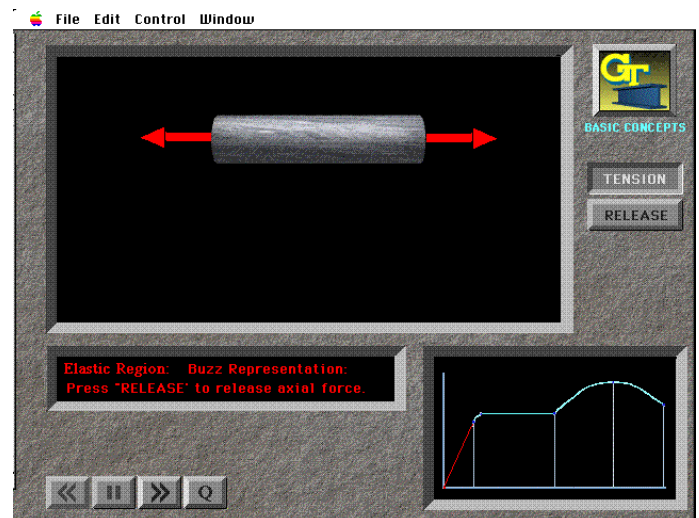


Fig. 4. Stress/Strain Module. Rod under tension.

planning for courseware production with multimedia software: the machine on which the courseware is produced, and the machine on which it is played back. For production, CPU speed is important in order to deal with the numerically intense modeling and rendering programs. For a Macintosh based system, a 68040 based machine (Quadra models) with a math co-processor or a powerful based system is essential. Since it is often necessary to run several development packages simultaneously, large amounts of RAM are required (16-20 Mb is a practical minimum). In addition, multimedia development packages and their output files are large in size, and this will require at least a 500 Mb disk system (a 1 Gb drive is better). Stereo playback with reasonable recording abilities is a desirable feature of the production machine.

Many of the multimedia development packages described above operate with several large windows on the display screen. As a result, development with a single 13 inch or smaller monitor can be cumbersome. A 16 inch or larger monitor is recommended. The preferred setup includes dual monitors which provide ample workspace for both scenes and scripts. When creating photo-realistic images, a monitor and video card with 8 bit color (256 different colors or shades of gray per pixel) can be limiting. The maximum color depth, 24 bits (approximately 16 million colors), can be problematic due to the large amounts of memory needed and the copious amounts of time required for rendering and play-back. Sixteen bit color (approximately 65 thousand colors) has proven to be a good compromise between image quality and processing speed. This more closely matches the limits of human color perception at about 300,000 shades. If 16 bit color is not available, 8 bit is recommended over 24 bit for most productions.

The computer on which the module will be played back must be considered during the development stage. Since most students cannot afford, nor have access to high-end systems, a module produced in 24 bit color will, most likely, not run well for the student. It is better to produce all modules to run smoothly within the performance constraints of a specified target play-back computer. All modules produced at Georgia Tech's Aerospace Engineering Multimedia Lab are designed to run well on a baseline Mac IIci (25 MHz 68030 CPU, 8Mb of memory, and an 8/16 bit, 14 inch monitor).

The large amounts of storage memory required for a multimedia module have made high-capacity removable devices a necessity for both production and playback. During production, these devices are the only practical means of transporting a module from one machine to another. Removable high capacity drives also allow multiple students to use the same machine since each student's work can be completely stored on the removable disk. Read-write optical drives have decreased in price and are currently a good alternative for production. These drives come in capacities from 128 Mb to 1.2 Gb. The 128 Mb drives have ample capacity for most programs and they are relatively inexpensive. Since students may not have access to large amounts of hard disk capacity, the modules must be distributed on high-capacity removable devices. For playback, modules will ultimately be transferred onto CD-ROM's. CD ROM's are an increasingly popular and inexpensive option on most new computer systems.

As mentioned above, the final product will often be copied to a CD-ROM for distribution. This requires making a master CD that can be sent to a production facility for mass reproduction. Before the final master is made, multiple test CD's are generally required to check playback speed, data structure, compatibility and final layout design. These test CD's can now be produced on personal CD recording equipment which is commercially available for little more than the cost of a computer system. These recording systems are not the same as those used in mass production centers, and each special blank CD-ROM costs about \$25. However, these recorded CD's can be played in any CD-ROM player and can be used for testing and limited distribution. They can also be used to make a master for mass production of a CD-ROM which can cost as little as \$2 each. CD-ROM recording equipment will become more important in future multimedia development.

Conclusions

The production of engineering courseware presents a host of problems which must be considered. First and foremost is money. Hardware, software, space, and personnel are all expensive. In addition, the time required to produce a working module is surprisingly long-on the order of 6-10 months for simple modules. In order to effectively produce engineering modules, the developer must have a working knowledge of all the software available. The numerous software packages that must be mastered present a formidable learning curve. Again, time becomes an issue. Finally, modules must be produced and tested with students in order to establish their value as a supplement to textbooks and lectures.

As with any new field of research, there are skeptics who dispute the credibility of engineering courseware development using multimedia. As a result, resistance to this field of academic work can mount before the final product can even be tested. Given the large amount of time and money that must be invested, educators must take a careful look at the advantages of developing multimedia engineering courseware. In particular, the advantages of a multimedia module over text books and lectures must be carefully weighed. The animation and interactive capabilities of multimedia add a dimension to the presentation of engineering concepts that previously did not exist. Properly designed multimedia materials can allow students learn faster and with more enthusiasm, and can let teachers cover topics that were previously very difficult or were avoided altogether.

Multimedia is here to stay. Its potential in the development of engineering courseware is only beginning to be understood. The development work conducted to date has provided an encouraging first impression of its value in college-level education, but detailed knowledge to help educators create new material and truly put multimedia to use has yet to be produced.

Acknowledgment

The authors would like to acknowledge the financial support of NSF and the SUCCEED coalition.

References

1. "Media Mania," Business Week, July 12, 1993, pp. 110-119.
2. Brown, E. "Will multimedia make college obsolete?," NewMedia, July 1993, pp. 18.
3. Naeim, F., "The San Andreas Fault System, A Multimedia Earthquake Encyclopedia," Proc. Computing in Civil and Building Engineering, Anaheim, CA June, 1993, pp. 367-369.
4. Deierlein, G. M. Valenzuela and R.N. White, "Multimedia and Case Studies in Structural Engineering," J. of Computer Applications in Eng. Educ., J. Wiley, 1993.
5. BeMent, S.L., "Student-Authored Courseware," 1990 ASEE Conf. Proc., Toronto, Canada, June 1990, pp. 1884-1886.
6. Gramoll, K.C., "Interactive Beam Analysis Program for Engineering Education," *Computer Applications in Engineering Education*, to appear in 1994.
7. Higuchi, H. and E.F. Spina, "Improving Undergraduate Engineering Education Through Scientific Visualization," 1993 ASEE Conf. Proc., Urbana-Champaign, June 1993, pp. 1581-1583.

ROBERT ABBANAT

Robert Abbanat is currently a graduate student at the Georgia Institute of Technology. In June of 1994, he expects to receive a Master of Science in Aerospace Engineering. While at Georgia Tech, he has worked as a Graduate Research Assistant in the Aerospace Engineering Multimedia Lab, where he has researched the development and production of engineering courseware using multimedia technologies. In 1992, Mr. Abbanat received a BS. in Physics with a minor in Computer Science from Boston College.

KURT GRAMOLL

Dr. Kurt Gramoll is currently an Assistant Professor in the School of Aerospace Engineering at the Georgia Institute of Technology. He graduated from the University of Utah with a BS in Civil and a MS in Mechanical Engineering. Later, after working at Thiokol, Inc., he received his Ph.D. in Engineering Science at Virginia Tech in 1988. Since then he has taught at Memphis State University and worked at the Japanese National Institute for Space and Astronautical Science. His technical research work involves filament winding of composite structures and fluid-structure interaction. He is also heavily involved in educational research, particularly in the development and use of multimedia programs in engineering.

JAMES CRAIG

Robert Abbanat is currently a graduate student at the Georgia Institute of Technology. In June of 1994, he expects to receive a Master of Science in Aerospace Engineering. While at Georgia Tech, he has worked as a Graduate Research Assistant in the Aerospace Engineering Multimedia Lab, where he has researched the development and production of engineering courseware using multimedia technologies. In 1992, Mr. Abbanat received a BS. in Physics with a minor in Computer Science from Boston College.