

Essays

Approaches to Cell Biology: Developing Educational Multimedia

Robin J. Heyden

Educational Multimedia Consultant, Wellesley, Massachusetts 02482

Submitted August 29, 2003; Accepted March 9, 2004
Monitoring Editor: Sarah C.R. Elgin

It's all around us. On television, at the movies, and on the evening news we see animations that take us inside a fatal wound, demonstrate how a spaceship achieves orbit, or reveal the intricacy of the immune response. Can these powerful new media tools be put to use in science research and teaching? Increasingly college and university faculty are engaged in the business of developing multimedia to augment their work. Science faculty have discovered the power of multimedia to simulate process or change over time, reveal multidimensional structure, and engage their students in useful, interactive scenarios to reinforce their understanding of science.

Kenneth C. Green, of The Campus Computing Project, reports that 62.5% of colleges and universities participating in the 2002 Campus Computing Survey offer at least one complete Web-based course. He further reports that well over half the participating institutions of higher education either have a campus Web portal (a large entry point-site to the World Wide Web, offering an array of services and links) now or have one under development (www.campuscomputing.net/). Peterson's online directory (www.petersons.com/distancelearning) lists some 1,100 institutions that provide online degree programs. Course management software providers Blackboard, eCollege, and WebCT report impressive profits and penetration at most colleges and universities throughout the country (Blackboard reports 38% second-quarter growth over 2002 performance, while eCollege reports 27% growth over the same period in 2002). These data point to an increased prevalence of multimedia on college campuses and indicate an elevated student expectation to use multimedia tools in their educational experience.

Educational research points to the effectiveness of using multimedia activities in instructional settings. The capacity for repetition, new methods for encouraging student exploration and discovery, assistance with complex and highly visual processes, and accommodation of a wider range of learn-

ing styles are all cited as key contributors to the effectiveness of these new tools (Brown, 2000; Cuban, 2001; Cates *et al.*, 2002; Kelly, 2003).

Building multimedia, however, is not a simple job. From my own biology media projects and working in educational publishing over the past 20 years, I can attest to the time-consuming and often frustrating nature of creating your own activities, animations, or simulations. If you have built a course Web site, constructed illustrated or animated Microsoft PowerPoint lecture slides, dabbled with animation production tools—or just wanted to do those things—this essay is intended for you.

We should begin with a few definitions. What is multimedia and what are the delivery options? As the word roots imply, multimedia combines text, images, video, sound, and animation to bring one cohesive experience, in our case, to the learner. A multimedia activity (or product) can be delivered to a student online (over the Internet) or on “hard” media (CD or DVD). It is a common mistake to confuse the delivery mechanism with the product itself. You might find yourself thinking, “I’d like to develop a CD to help my students understand cell division,” when what you should be thinking is, “I want to develop a multimedia activity to help my students understand cell division and I will *deliver* it to them on a CD.”

The scale of multimedia development varies widely. There are very small projects (for instance, a presentation consisting of a stepwise progression of still graphics in Microsoft PowerPoint) and very large projects (for example, a library of 3D visualizations to support your research) and everything in between. Whether your project is small or large, the principles and guidelines in this essay should help to get your project off to a good start and make the most of your efforts.

This essay begins with a survey of the most commonly used multimedia production software, comparing their advantages and disadvantages. Following that, the essay outlines the steps for getting started, suggests a typical product development pathway, and discusses the various roles (talents and skills) required to complete a multimedia project. A short section on how to effectively run a team and avoid common problems follows and then the

DOI: 10.1187/cbe.03-08-0009

Corresponding author. E-mail address: rheyden@comcast.net.

Software Tools (in order of increasing complexity)	Description	Advantages	Disadvantages
QuickTime [From Apple. Latest version = QuickTime Pro Note: QuickTime Pro is the authoring environment, QT player is the free viewer for user playback] www.apple.com/quicktime	Authoring tool for editing, building, modifying, and displaying video. Can also create movies out of still images or layered images. (Note QuickTime VR is a type of QT movie that allows a user to view a 3D object from as many angles as specified, using the mouse to control the perspective.)	- Very easy to use and inexpensive - QT movies can be imported into Flash or Director for further manipulation - Free player plug-ins for viewing completed movies	- Limited to making movies - Student interaction and control is limited to the movie control panel (pause, play, rewind)
Flash [From Macromedia. Latest version = Flash MX. Note: Flash MX is the authoring environment for those who plan to create animations, Flash 6 player is the free plug-in for user playback] www.macromedia.com	An authoring tool for creating animations, designs, and interfaces for the web. Emphasis is on small file size for quick downloading.	- Vector-based graphics (so animations can easily be resized) - Relatively easy to learn - Great for creating quick-to-download web content - Free player plug-ins for viewing completed animations	- Has its own scripting language (ActionScript) but it's not as strong and capable of handling complexity as Director. - Not good for creating 3D.
Director/Shockwave [From Macromedia. Latest version = Director MX. Note: Director is the authoring environment for those who plan to create animations, Shockwave is the free player for user playback] www.macromedia.com	An authoring tool for creating graphically rich, interactive multimedia for CD, DVD or the internet. Emphasis is on complex programming and handling larger images and files	- Very powerful, includes its own scripting language (Lingo) which can be used to create real-time 3D - Free player plug-ins for viewing completed animations	- A bit more complicated to use, harder to learn, not as good for delivering quick and responsive web material
JavaScript [From Netscape Communications] Java [From Sun Microsystems]	These are cross-platform languages designed to create animations and interactive features on web pages. Java is a general purpose programming language. JavaScript, designed to work in a web browser, is a scripting language, which is a bit easier to learn.	- Powerful tools - Can build highly interactive activities	- Requires the talents of a programmer - Can encounter browser and end-user platform compatibility issues with JavaScript
LightWave [From Newtek] www.newtek.com This is just one of many available 3D animation programs (others include: 3D Studiomax, Cinema 4D, Maya)	3D animation package. LightWave-created scenes or objects can be exported as Shockwave files, thus combining the content creation tool with the interactive functionality of Director.	- Very powerful - Widely used by the gaming and movie industry for special effects	- Difficult to learn - Expensive to purchase - Requires the talents of a programmer

Figure 1. Multimedia software tools.

essay concludes with an examination of the importance of evaluation.

TOOLS AND SAMPLES

Figure 1 briefly summarizes the software tools most commonly used to create multimedia experiences and activities. In general, web pages are built in HTML (Hypertext Markup Language). HTML works in the background to arrange

the web page and display the text to be read. Animations, activities, movies, or simulations, built in one or more of the tools above, are inserted into the web page and function independently from the HTML.

All of these tools are relatively challenging to master but some are more difficult than others. Learning to put together (or program) a simple animation in Macromedia Flash, for instance, would take more time than learning to use your e-mailing software, but it would be faster than learning to

	Small Project Samples	Large Project Samples
Built using QuickTime	http://chemistry.beloit.edu/Rain/index.html (layered movies showing titration curves, part of the Chemical Connections program)	http://panax.web1000.com/qtvr.html (QTVR panoramic images of European cities)
Built using Flash	http://www.cellbioed.org/resources/resources.html (short topic animations provided by the CBE) http://biocomp.chem.unb.ca:8080/GD/flashware/chem-samples/chem-samples.html (series of small interactives for the college-level organic chemistry course) http://www.hhmi.org/biointeractive/clocks/animations.html (simple animation of the Drosophila molecular clock) http://www.bml.psy.ruhr-uni-bochum.de/Demos/BMLwalker.html (simple animation of a human figure walking)	http://www.brainpop.com/ (Winner of the Flash Forward 2003 Festival Award, Education category. Web site full of animated movies for K-12 students) http://www.moma.org/russian/ (an interactive online exhibition designed for the Museum of Modern Art displaying works of the Russian avant-garde)
Built using Director	http://web.mit.edu/8.02T/www/802TEAL3D/ (Studio physics project which includes 3D animation of electromagnetic forces for teaching first-year physics)	http://www.adam.com/demos/maindev.asp (detailed anatomical products for medical students and professionals) http://www.hhmi.org/biointeractive/vlabs/index.html (a series of highly interactive, virtual science and technology labs produced by Howard Hughes Medical Institute)
Built using LightWave	http://www.hhmi.org/biointeractive/gender/index.html (short QT animations on sex determination built with LightWave and AfterEffects)	http://www.sybilmedia.com/ (a 20-minute 3D movie created for the Houston Museum of Natural Sciences on extreme natural forces)

Figure 2. Sample projects.

use Macromedia Director or LightWave or teaching yourself how to program in Java.

If your plans include creating a few simple animations on your own, Flash is a good place to begin. If you are planning a larger-scale project with complex visualizations, then you should consider more serious tools and hiring talented students or others to do the programming/scripting.

A useful rule of thumb is that the authoring challenge is proportional to the complexity of the product you plan to build. For example, you know that writing a letter in Microsoft Word takes much less time to create than using the same program to build a complex table with inserted links and graphics. If your plan is to create three or four simple animations in Flash, it might take you a day to learn how to use Flash and another week or two to build those simple animations. If you plan to develop an animation for every lecture in your introductory class and post them on a Web site for your students, you will most likely be working for months.

Media projects come in all shapes and sizes. To get a feel for the range, the links in Figure 2 will lead you to representative

projects. These examples demonstrate the range of software tool capability as well as the tremendous variety in project complexity.

GETTING STARTED

The beginning of any multimedia project is no different from the beginning of any complex endeavor—you must first determine what you intend to do and then plan the work ahead. Productive time spent here will reap rewards later. As a house-painter friend once told me, painting a house is 90% preparation and 10% painting. The same goes for multimedia construction.

It is best to begin with a clear understanding of your intended audience and the needs you hope your project will satisfy. In the case of multimedia development, knowing your audience includes knowing about their computers. You must determine if the computers your students will use can meet the technical requirements of your planned project. Competent engineering assistance is required during this phase



Figure 3. A typical project development pathway.

of your planning. Technical requirement considerations include computing power, memory, screen size and resolution, likely Internet connection speeds, installed operating system and version (i.e., Windows 98, Windows 2000, Windows NT, Macintosh and its various versions), preferred browser and version (i.e., Netscape or Microsoft Internet Explorer and their various releases), and appropriate plug-ins. As described in Figure 1, animations created with most of today's multimedia authoring programs require end users to install a "plug-in" on their computer in order to view them. Plug-ins are small files that instruct the computer how to display objects created using the parent program. Generally available without fee, plug-ins are specific to each operating system. You need to know if your audience will have (or can get) access to the plug-ins required for viewing objects built in the software you intend to use.

Similarly, it is helpful to be familiar with your audience's background. Why will they use your project? What existing knowledge do they have? and How will they apply lessons learned from your project? This information will help you tailor your plans to meet their needs.

Once you assess your audience and determine your plan you can begin to take stock of the potential resources available to you. Possible sources of talent include graduate students, interested undergraduates, and students in other departments (computer science, graphic arts, business)—any of whom might have the skills you require.

Before work begins, take a critical look at the costs. Examine each project element, estimate how long it will take to build, and multiply that figure by the hourly rates of the people involved. You will also need to estimate the hardware and software costs, equipment purchases, permissions fees, research fees, and any overhead costs. It is advisable to then add in a reasonable "cushion" (some use 10% of the total figure) for the unexpected.

As for schedule, in addition to the time you estimate for building be sure to allow time at the end for quality assurance, compatibility testing (with the various operating systems and browser versions), and the inevitable problems that will arise.

It is also recommended that you give some consideration to how you will know when the project is complete. This sounds like a trivial question but an embarrassingly large number of media projects suffer from no one asking it. When you visualize the end point, what does it look like? How will the project behave? What will be the benefit to the student or the colleague using it? How will you assess the project's ultimate effectiveness? The time you spend thoughtfully considering these questions will greatly benefit you, your project, and your students.

PRODUCT DEVELOPMENT: PATHS AND FUNCTIONS

One way to plan your project's development is to consider the steps required to build it. A flowchart of your development pathway can be a useful tool. A simple example is shown in

Figure 3. Any individual step in the path might have multiple iterations or component parts but there should be a stepwise, logical flow from idea to finished piece. As you lay out your pathway, the skills and talents (people) required to accomplish each step in the path become more apparent. A pathway chart like this will also help you to build a project schedule.

Whether large or small, most media teams will include members who play the following functions: programmer, designer, artist, engineer, editor, and project manager. Small teams might have members serving multiple functions, but all functions must be explicitly addressed.

Programmers. Most educational science multimedia is highly graphical, consisting of visual elements that move, change, or are acted upon by the student. The action of these visual elements is scripted (or "programmed") by a person who might call him- or herself a programmer, a software engineer, or an animator (the term "animator" is used when the programming involves mostly visual elements). Software engineers typically specialize in one particular software tool or environment. For instance, if you plan to use Flash to build interactive animations, then you will need the services of a Flash programmer. Don't be tempted by anyone who claims that they can "teach themselves" the software tool you intend to use. You will need expertise here that comes from experience. If you are developing 3D animations, the skills of a highly trained 3D animation specialist will be required.

Designers. A well-constructed Web site or software program's graphical user interface (or a "GUI" as it's called in the business) is the work of a designer. What we might not realize is that there are two important skill sets required to create a good interface design. First, there is the *graphical design*, or how a product looks. Graphical design encompasses such elements as color, type treatment, logos, and illustrations. But there is also *information design*, which involves navigation and the user's path through the product or program. Unfortunately, information design is frequently overlooked or left unattended. Make sure that you have found a designer who is competent in both areas. While it is wise to begin thinking about design early in the project, you will need to have enough of your product in a semifinished form to give the designer material with which to work. So, for example, if your product consists of four categories of objects or activities, you should have at least one sample from each category completed (or in substantial draft form) so that the designer can fully understand your design needs and offer proposed solutions.

Artists. If your media project has a visual component you will most likely need an artist. Professional scientific illustrators can make significant contributions to your project. Because they have science training, you will not need to explain the basic science to them. You should also look for artists who work with computer tools—that is, artists who use Adobe Illustrator and Adobe Photoshop rather than watercolors and pastels

to create their work. It is also possible to find trained artists who can work in Flash or Director, combining the art and programming into one step (animation). If you have many illustrations in your project, you might consider hiring an art director; someone to oversee the art program, create storyboards, and manage the artist(s) working with you.

Infrastructure/Engineer. If your program will be delivered online, it will be housed on a server, most likely on your campus. If your program will be delivered on a CD or a DVD, there will be compression (making sure your file sizes are small enough to fit on a CD or DVD) and compatibility issues. Regardless of delivery method, you will need the help and advice of an engineer. Most likely this assistance can come from your Technical Support Group or Instructional Technology department on campus. It is critical to include these people in your planning from the outset. They will help you with issues of delivery, software limitations, compatibility, compression, page weights, server calls, troubleshooting, and long-term maintenance. They can also help you set up an online development environment in which you can post draft stages of your program for contributors or reviewers at distant locations to preview.

Editor. Just as with print products, electronic products require careful editing. Editors will help you to clarify your instructional messages, check your accuracy and consistency, make certain you have accomplished your instructional goals, and look for opportunities for improvement. Editors do much more than correct your grammar and spelling (although they do that too)—they stand in for the students and help you predict places where your end user will falter or fail to comprehend.

Project Manager. Every complex project needs a leader and a whip-cracker to bring it all together. Project managers are the keeper of the budget, the schedule, and the overall process. They move manuscript, storyboards, and partially completed segments along from one spot to the next, keeping everyone informed as they go. Look for someone who is organized, confident, and enjoys making order out of chaos.

TEAMWORK, OR GETTING A TEAM TO WORK

With any team—small or large—there can be difficulties. Personality conflicts, problems keeping everyone in harness, unequal contributions, distractions, delays, and setbacks can demoralize the team. The key, as team leader, is to keep everyone well informed, happy, and productive. Here are a few suggestions that will make any team effort function more smoothly.

- It is important to recognize that everyone has a different perspective. Each specialist comes to the team with a different background and expertise. Make sure that everyone on the team understands all of the roles and the valuable contributions made by their colleagues. People can only respect what they understand.
- Get to know the language of each specialist. Beware of jargon, acronyms, industry slang, or science shorthand that might exclude or baffle people. Establish a model yourself by always communicating clearly and using jargon only when necessary and, in those cases, explain it clearly.

- Establish team goals and standards. A team functions best when everyone clearly understands the end goals and the quality standards. Work with your team early to establish these and then find ways to regularly remind everyone of the shared principles.
- Communicate clearly and often. You will be staggered by how many times you must repeat the project goals, explain the needs of the intended audience, remind people of the plan, or reiterate the deadlines. Use the vehicles of regular team meetings (in person or via conference if team members are in different locations), progress charts, weekly update reports, project management software (such as Microsoft Project), or project Web sites to keep everyone in touch. Prepare yourself for the need to communicate with team members regularly.

CLASSIC CHALLENGES IN PRODUCT DEVELOPMENT

Scientists are very familiar with the classic challenges that arise in their research. While those, or challenges like them, will certainly show up in this work, multimedia product development poses a few unique challenges that warrant your attention.

Accuracy. If you are constructing animations, simulations, or movies, you will need to pay very close attention to details. Accuracy issues are much more complex and insidious when data or objects are moving, changing with time, or altering due to student input. Familiarize yourself with all the new ways your material could be considered inaccurate and devise methods for rooting out those errors.

Realism versus Teaching Value. Educational products are designed to teach, and in many cases, that might mean sacrificing realism for teaching value. In other words, beware of places where the teaching point might be obscured by adding excessive detail in the name of accuracy. Always ask yourself, “what is the main point of this activity (illustration, animation, or simulation)?” and let the answer serve as your guide to what detail to include or exclude.

Too Much Glitz. Many first time media developers fall into the trap of doing things because they can be done. It isn’t always better to include glitz, pop, or gee-whiz effects with easily distractible students. Simplest is often best.

Multiple Representations. With multimedia we now have the wonderful ability to show multiple and simultaneously occurring events or processes. But that isn’t always pedagogically most effective. For example, it is now entirely possible to show a beautiful animation of a beating human heart (including the activity of the heart’s intrinsic conduction system, the opening and closing valves, and the flow of blood from chamber to chamber), next to an animated ECG, correlated with the heart sounds and pacemaker activity. But a typical student would be lost when presented with all of that information simultaneously. Careful thought must be given to the main point of the activity. If multiple representations do support the learning goals, design the experience so

that the student can navigate efficiently and make selective comparisons.

“Open-Endedness.” Media developers struggle over how open-ended to make their products. While we want students to be engaged, to have some stake in the outcome, and to make choices, we don’t want them to get lost or frustrated. With few exceptions (mathematically driven simulations for instance), multimedia activities offer a limited amount of student interaction. As an author, you must decide how much latitude to provide within those limitations. A useful generalization is to offer enough direction to accomplish the objective but not so much that the student becomes passive. Another is to offer enough choices to engage the students but not so many as to overwhelm them. You will also want to consider how the activity can be applied within the larger instructional context. Ultimately, the appropriate degree of “open-endedness” for the student should depend on how the multimedia experience is used and combined with other instructional methods and tools.

Words versus Pictures. The computer screen is not the best place for serious reading; no one likes to read long passages this way. So resist the temptation to launch long explanations and save your exposition for print.

EVALUATION

Once a media project is completed, the natural tendency is to leave it and quickly move on to the next project or challenge. But your work is not complete until you have carefully evaluated the product you created. Did it work? Was your original goal met? What have you learned that you can apply to your next project or to the improvement of this one?

It is strongly recommended that you plan your evaluation path up front, at the beginning of the project. Decide on your test bed (Who will evaluate your project?); determine if the evaluation will be *formative* in nature (providing you feedback as you build) or *summative* (providing you feedback on the completed work); decide on the questions you will ask and how you will evaluate the answers. By planning these elements up front, you will be in an excellent position, at project’s end, to answer the question—Did it work?

You can do the evaluation yourself, you can designate a person on your team to conduct the evaluation, or you can bring in outside expertise. There are evaluation experts who can help you to construct an authentic and objective evaluation and help you translate the feedback into actionable items and improvements in your final product plan.

Multimedia product development can be a rich and rewarding experience. The tremendous power of the Internet and new development software tools combine to give scientists and teachers a powerfully creative outlet for our fascination with visual models. If you thoughtfully plan, recruit talent, develop, and then evaluate your work, you are sure to build excellent multimedia that will benefit your students and colleagues.

Resources

Information about Flash and Director. [http://www.macromedia.com/downloads/\(downloads and information about Flash and Director\)](http://www.macromedia.com/downloads/(downloads and information about Flash and Director))

[http://www.macromedia.com/support/flash/\(Flash support from Macromedia\)](http://www.macromedia.com/support/flash/(Flash support from Macromedia))
www.wireframe.co.za
www.vectorlounge.com/04_amsterdam/jam/wireframe.html
www.flashforward2002.com (examples of interesting ideas executed in Flash)
<http://www.virtual-fx.net/vfx/> (Flash tutorials)

Scientific Illustrators/Artists. [http://www.gnsi.org/\(The Guild of Natural Science Illustrators\)](http://www.gnsi.org/(The Guild of Natural Science Illustrators))
[http://www.ami.org/\(The Association of Medical Illustrators\)](http://www.ami.org/(The Association of Medical Illustrators))
<http://huntbot.andrew.cmu.edu/ASBA/ASBotArtists.html> (American Society of Botanical Illustrators)
[http://www.iaaa.org/\(International Association of Astronomical Artists\)](http://www.iaaa.org/(International Association of Astronomical Artists))
[http://www.gag.org/\(Graphic Arts Guild\)](http://www.gag.org/(Graphic Arts Guild))

Using Technology in Teaching. <http://www.nap.edu/html/howpeople1/ch9.html>

An excerpt from an NRC publication called *How People Learn: Brain, Mind, Experience, and School*. This link takes you to the chapter on “Technology to Support Learning.”

<http://www.stemnet.nf.ca/~jscaplen/integration/english/prep2.html>

An easy-to-digest, bulleted list of classroom management suggestions assembled by Jane Scaplen.

<http://www.webteacher.org/windows.html>
 A useful tutorial site on using the Web in the classroom, sponsored by TechCorps.

<http://www.electronic-school.com/>
 An online magazine about school technology sponsored by the *American School Board Journal*.

<http://www.technologygrantnews.com/>
 An online newsletter, published four times a year, that brings you timely updates on available technology grants.

<http://www.masie.com>
 A Web site and online monthly newsletter that discusses key issues at the intersection of learning and technology.

<http://www.usingexploringlife.com>
 An informational Web site about my latest project—a high-school text and Web program.

REFERENCES

- Brown, J. S. (2000). Growing up digital; How the Web changes work, education, and the ways people learn. *Change*, March/April.
- Campus Computing Report. <http://www.campuscomputing.net/pdf/2002-CCP.pdf>
- Cates, W., Price, B., and Bodzin, A. (2002). Implementing technology-rich curricular materials; Findings from the exploring life project. *Comput. Schools* 20(1/2).
- Cuban, L. (2001). *Oversold and Underused: Computers in the Classroom*. Cambridge, MA: Harvard University Press.
- Kelly, H. (2003). Education for tomorrow needs innovation today. *Carnegie Rep. Spring*, 44–45.