

Exploiting and Integrating Multimedia

by Phillip Kerman

Multimedia can teach, inspire, excite, and communicate in many ways. But how can a technical communicator tap into this power? There are “new media” tools and methods to exploit this medium—from simple solutions using interactive Adobe Acrobat files, through Flash animation, to Director and Authorware applications for CDROM and the web.

Everything’s easy—it just takes time. Though this statement may seem simplistic, the tools to create multimedia are not particularly difficult. You’ll see successes almost too early. People often mistake “easy” for “fast”—and in the case of multimedia, everything takes longer that you think.

Tools:

Authoring tools are not to be mistaken with production or *creation* tools. In the case of creation tools (like Photoshop, Illustrator, AfterEffects, etc.) the output is raw media. Static images, audio files, video files... even linear animations—it’s all just media. *Assembly* tools (that is, authoring tools like Director, Authorware, and even Flash) don’t really create anything from scratch—rather they assemble media created previously. “I put pictures on the computer and make them move” is the salient job description from a multi-media author.

This table suggests a range of authoring tools, placed in order based on decreasing ease-of-use but increasing power.

Acrobat (with ‘links’)	Flash or Director (linear animations)	Authorware	Director’s ‘Lingo’
Cheap. Easy. <i>Can</i> including movie clips, but ‘interactivity’ is limited to clicking links.	<i>Can</i> do both animation and interactivity. However, they’re good choices for their built-in animation capabilities. (Flash is less expensive.)	Highly interactive applications are within the reach of any technical communicator (flowchart metaphor).	Full-fledged programming language (for programmers, but not limited).

Ways to communicate:

People learn in different ways... and different methods of communication are effective in different ways.

Method	Appropriate situations	Effectiveness/Appeal
Text	Existing material already in text form—might as well get more mileage from previous investment in work.	Some people respond very well to text. However, on-screen text is not easy to read. Only a fraction of what can be printed on paper will look clear on screen. The bottom line: people don’t read.
Pictures / Graphics	Existing photographs. Photos are good for unique objects... graphics are good for generic objects. For example, you could use a photo of a particular fish... but use a graphic to represent sea-life. Photos, often, are taken very literally.	1 picture = 1000 words. See “people don’t read” (above). Appeals to visual learners.
Animation	Subject is time-based. Sequential information like processes. Also, theoretical models can exploit animation. User may want to watch the <u>same</u> animation repeatedly.	Investment to create an animation can be extensive. Realize, there are different levels of animation (from 3D with audio to static images that layer). Animation has a lot of un-tapped power—use it!
Interactivity	Remember, there are many <i>types</i> of interactivity. Text entry, “hot spots”, drag & drop, etc. Consider the investment when subject matter is <i>deep</i> enough to entertain infinite variations—that is, detailed enough to keep the user occupied.	You can appeal to multiple senses: auditory, tactile, visual. Also, the computer is patient—some people may spend more time interacting than others.

Efficiency Techniques:

Prototyping involves the final media form as soon as possible. This way all people involved can better communicate and judge the final application. Templates can automate your work... but the trick is to make them efficient for you, the author, while remaining fresh and unique-looking for the user.

Resources:

Related papers I’ve delivered:

Multimedia Production Basics: www.teleport.com/~phillip/toronto/nm98_226.pdf

Building Templates for Large Projects –Rapid Prototyping www.teleport.com/~phillip/toronto/nm98_317.pdf

Find your local *Macromedia User Group*: www.macromedia.com/support/programs/usergroups/

Awesome site for learning multimedia techniques: www.webmonkey.com

Software: www.macromedia.com www.adobe.com www.equilibrium.com

Credits and References:

Sites/Companies referred to: www.io360.com, www.seagrants.orst.edu, www.newi.com, www.wagged.com, www.hp.com

Graphics: Diana Bauer (dbauer@teleport.com), Brandon Blank (BBBadger@aol.com), Christian Moore (webputty@choice.net).

Glossary of terms as applied to multimedia:

anti-aliasing—When edges of images or text is blurred and combined with the background resulting in smooth edges—no “jaggies” or “staircases”.

batch processing—Applying a sequence of processing steps to a list of files.

code data separation—Programming technique keeping code (or programming) separate from data (or media).

color palettes—A table with 256 spaces representing all the colors used in an image. Each pixel uses one color from the palette. The 256 colors in a particular palette may be anything—so an image with lots of shades of red, for instance, can utilize a palette with lots of reds. This way images can look good, remain small in size, and work with machines using limited video displays. Switching palettes is easy, but a strategy needs to be applied to avoid palette “flashes” during the switch.

cross platform—Commonly refers to applications for both Mac and Windows. But now includes Mac/PowerMac/Windows3.1/Windows95-NT—or more. Also, the term “*cross browser*” is becoming popular.

dithering—Algorithm to reduce colors by using dots and blurs to create the appearance of more colors.

dynamic—opposite of static. Referring to portions changing at *runtime*.

Lingo—scripting and object oriented programming language for Macromedia’s Director.

localization—converting a finished project to another language or culture. The trick here is to make the conversion seamless.

mask—a graphic (sometimes simply black and white) used to block portions of another graphic.

matted video—when motion portion includes parts of the surrounding static background it is said to be matted.

models and templates—an adaptable code segments which can be utilized many times throughout a project.

onion skinning—animation technique where you can see through to other frames while editing one frame.

placeholder—a temporary media file “holding the place” for the finished media. Work can occur simultaneously.

plug-ins—When software publishers provide third-party programmers with the ability to create custom extensions to their products (like Visual Basic “VBXs”, Macromedia “Xtras”, or Adobe Plug-ins by a company like Extensis—in Portland).

render—processing graphics or animations before incorporation into a project—once rendered, changes may not occur. Less *dynamic*, but better performance.

prototype—common way to confirm agreement among a design team.

registration—assurance that images are “lined up” properly.

runtime—At the time the user “watches” a project. Effects that occur at runtime require the user’s machine to do the work. Can also refer to the portion of an executable necessary to view proprietary files.

scripting—another word for programming, usually less technical.

sprite animation—static images moved around on screen.

target machine—specification of what system requirements.

treatment—a description of how a project will achieve its objectives.

user interface—the way the user interacts with your software... the screen, the buttons, the way they use a mouse or keyboard.

tweening—When “key frames” of an animation are created and then in-between frames are filled in (either by hand or with software).

video compression—Algorithms to reduce file size. Video files can be compressed as they’re captured or after editing... then decompressed at *runtime*.

Xtras—proprietary (though open) architecture for custom “plug-ins” for Macromedia products.