Multimedia for Authoring Motion Pictures

Alan J Rosenthal
flaps@dgp.utoronto.ca
(416) 923 6641 ext. 2279

Ronald M. Baecker
baecker@dgp.utoronto.ca
(416) 978 6983

Dynamic Graphics Project
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 1A4
Fax: (416) 978 5184

Abstract

In this article we present a computer program called "MAD", Movie Authoring and Design. Whereas most computer software for filmmaking focuses on the editing and post-production stage, MAD is designed specifically to support the authoring stage, and may be used before any film footage is shot. It assists in the process of developing and refining the concept for a movie by supporting the intrinsically hierarchical nature of movies; it supports a top-down design approach as well as a bottom-up implementation approach. MAD allows the user to keep script, storyboards, sounds, and digitized video clips together in a single document. Visualizing the final result is assisted by the "play" feature, which allows an approximation to the final film to be played on the author's workstation at any time. The accuracy of this approximation increases as additional script, timing information, and other data is added to the movie.

Keywords: Multimedia, Motion Pictures, Authoring Tools, Interactive Systems.

Introduction

The goal of this work is to enhance significantly the ease and creativity with which filmmakers are able to author and create motion pictures. Inspiration for this work has been drawn in part from recent advances in technology for writing documents, designing software, and creating music.

Written documents

Probably the most common use of computers today is the creation and editing of "documents", such as papers, memos, notes, and books. "Word processors" allow editing of documents, so that subsequent drafts do not require complete re-typing but only the typing of the changes to the document. They also create new ways of thinking by allowing users to write documents in an arbitrary sequence. Word processors make it easy to navigate around a document which gradually coalesces. The user can add additional text at any point in the document. It is also common to use "place-holders", where the user types strings such as "???" or "more...", intended to be filled in later. While the draft of the document gradually moves towards its final form, the user can print the document at any time. The draft document is printed with all section headings and place-holder text and thus forms an approximation of the final form of the document.

"Outline processors" (such as "MORE", Symantec, 1990) augment this with explicit support for a hierarchical structure. A book consists of chapters; a chapter may consist of sections; and so on. A presentation may consist of an introduction, a series of general topics, and a conclusion. Each particular topic may consist of subtopics, and so on. An outline processor provides facilities for creating and manipulating the hierarchy as well as the text in it. The detail of certain portions of the structure can be suppressed while one's editing attention is focused elsewhere. The user can work in a "top-down" approach or in a "bottom-up" approach. For some, an outline processor is the application program of choice for the creation and editing of certain kinds of structured documents or plans. Again, an outline can be printed at any time even though it is incomplete.

Computer software

Modern computer programming practice illustrates the utility of a hierarchical structure. Skilled computer programmers use a steep structure in which software components are broken down into subcomponents, subsubcomponents, and so on, to many levels of depth. Computer programmers find a good structure to be essential to performing their craft.

Despite an apparently adequate expression of computer program structure in computer programming languages themselves, some programmers pursue more advanced tools for manipulating or at least displaying the structure of the computer program or its associated data. Some of these involve sophisticated layout and display algorithms (e.g. Graham and Cordy, 1990). It is important to note that these representations are often not equivalent to the original information; unimportant information may be omitted and additional information may be synthesized (e.g. Baecker and Marcus, 1990). Increasing use is being made of visual representations, which are quite valuable because a different modality can cause a fresh interpretation of information. The value of visual representation for computer programs and other structured objects is discussed at length by Martin and McClure (1985), who present a variety of kinds of visual representations and extensive advice on their use.

When applied to computer software, these techniques are known collectively as *software visualization.* There is a substantial body of work in this area; over one hundred references may be found in Price, Baecker, and Small (1993).

Music

Many segments of the music industry have been transformed over the past decade by the advent of synthesizers and mixing boards controllable by MIDI commands (Loy, 1985). Now many composers will create their works using synthesizers and computers. A computer can store sufficient information to be able to create a facsimile of non-digital music by controlling sampling synthesizers; this facsimile can be suitable for many exploratory purposes. Computers can also be used to create a list of audio mixing commands to assemble the final version of the recorded music. In both cases the user realizes the usual advantages of computer technology; an error in mixing can be repaired without re-mixing the entire piece of music. In the synthesizer case, an individual can hear an approximation to what a larger ensemble of musicians would sound like when playing this music, which provides a "visualization" tool analogous to the early printing of draft word processor documents.

One notable result of the advent of MIDI is that it provides a novel "external" form of music. As with any other creative medium, music can be presented (performed) but is difficult to describe by other means. Nevertheless, musicians require methods of communicating about music. The chief method of describing music other than by presenting it is to produce sheet music, a written representation of music. MIDI has added a new external form of music. Certain kinds of musical ideas and proposed musical recordings can now be communicated via floppy disk.

Guide to this paper

We begin with a little background on the use of computers for motion picture and multimedia production. We then begin our discussion of MAD by presenting some design goals. Next, we describe our prototype implementation, still under development but already in use. One experience of using the system is described. Finally, we present some future directions and planned extensions to these ideas.

Uses of Computers for Motion Picture and Multimedia Production

The motion picture production process can be divided into "pre-production", shooting, and "post-production". Pre-production includes endeavours such as scriptwriting, storyboard drawing, set design, location scouting, and the hiring of actors. Post-production includes the selection from alternate "takes", assembly, editing, sound mixing, and adding visual effects.

Pre-production

MacroMedia Director (MacroMedia, 1991) is an integrated system for designing a visual sequence. It is a complex scripting environment in which the elements can be arbitrary images, sounds, or externally-controlled devices. The user creates a "score" which specifies which "cast members" (images) appear at what times, and where. The score also specifies at what point sounds are started and stopped (cut off). Arbitrary computer program text can be associated with cast members or with "frames" (sequence points), or it can be global to the movie. This achieves a very high degree of flexibility in the presentation.

Unfortunately, there are many drawbacks to Director. Its major flaw with respect to the authoring of a film is its lack of explicit movie structure. With Director, we can lose the fundamental advantage that computer technology can bring to an application area, that of easy revision. Early choices influence subsequent work so profoundly that it is often not worth attempting to make major changes to a Director "movie"; the cause of a given visual event may be distributed throughout the Director "movie" structure. This issue of structure in Director is discussed more fully by Hardman, van Rossum, and Bulterman (1993).

Director also suffers from a number of minor shortfalls which render it unsuitable for traditional films, such as extreme imprecision of sound synchronization. A Director script whose sound is roughly synchronized on one computer may produce very bad results on another.

Post-production

In recent years there have been several computer post-production systems capable of supporting the formulation of the final edit, such as The Avid Media Composer (Avid, 1993). With Avid, a filmmaker digitizes all of their footage from videotape onto a large storage volume. Avid then allows most of the usual film editing operations; using Avid to edit videotape feels more like film editing than video editing and has many of the advantages of the film medium. The user can form film clips and arrange them in bins. The digitization is frame-accurate, so final decisions can be made regarding cuts. When the visual quality of an individual digitized frame is insufficient, Avid can manipulate a computer-controlled VCR to display the frame from videotape on a video-editing monitor. The user can edit the entire film using Avid, and then use Avid to create an "edit decision list", which contains sufficient information for video editing equipment to assemble the final form of the film.

A system such as The Avid Media Composer addresses solely the problems of editing and post-production. Obviously these are crucial parts of the filmmaking process, but they are only parts. Use of Avid begins only after all video footage has been shot; shooting often begins only after a completed script has been decided upon.

Multimedia

Most modern computers can present a variety of images and sounds, not just text. Many can produce detailed colour images; many can produce good musical sounds; some can present full-motion video. An increasing number of media are able to be stored electronically or digitally, and many modern electronic devices are designed to be able to be controlled by computers. "Multimedia" systems attempt to bring these various modalities to the computer user.

Research in this area tends to address either the issue of how to use and organize or sequence all of the information (e.g. Pea, 1991), or how to increase access to media and devices (Woolsey, 1991). With respect to the current work, organizing the information is of greater interest, since video and imaging equipment has been computer-controllable for quite some time. Davenport, Smith, and Pincever (1991) propose the use of film terminology and concepts for the organizing of video clips. However, their project quite clearly veers away from the goals of traditional filmmaking. It is not capable of performing at professional filmmaking standards because it treats shots as atomic: crucial editing operations are unavailable, and sound cannot be manipulated independently. Generally, these projects share with MacroMedia Director an orientation towards a non-filmmaking audience.

Design Goals For Mad

Supporting multiple data types in an integrated fashion

In the authoring of motion pictures, various documents and sketches are produced. There may be a script in the classic form as well as a separate storyboard script. The properties and uses of these various items are discussed in detail by Katz (1991). Production information may be tracked by handwriting on one particular printed copy of the script which may become obsolete but be retained because it has the production information on it. The director may also produce a version of the script which focuses on direction. Others may do what a computer programmer might think of as "sorting" the script, to organize it based on shooting schedule, props required, or actors required for shooting.

While working on a script on the computer, various elements could be available. The Macintosh computer supports various "data types" in addition to text such as pictures, digitized video clips, and sounds. We believe that all of this data should be manipulated in an integrated fashion where possible, and a good movie authoring tool would allow the user to attach these things to the script. Alternatively stated, a good movie authoring tool would allow the user to attach the script to these various data types.

Supporting hierarchical structure: top-down design, bottom-up implementation

A film has a complex structure. One of the tasks involved in authoring is moving around in that structure, as discussed earlier with respect to document processing. In traditional filmmaking, a substantial amount of time and effort is devoted to organization. Similarly, any film author has overall goals, and there is a structure to the film; there is more than just the images and sounds. An example of an overall goal is the highest-level organization of the topics discussed by a film.

It is often beneficial to think in terms of a hierarchical structure. There are different methods of breaking down a film, but a film may consist of acts, which may consist of scenes, which may consist of shots, which may consist of individual movie frames. We believe that film authoring is often done in a "top-down" fashion, in which the overall structure of the film (e.g. acts) is decided upon first, and only after the breakdown into individual scenes are individual shots considered. An early preoccupation with details can focus on a bottom-up approach; support for a hierarchical approach would allow users to organize their thoughts in a structured manner. Of course, experienced filmmakers achieve this structure of thought already, through the depths of their experience, and attempt to have a complete script written before becoming entangled with lower-level details.

The experience from computer software design tells us that an exclusively top-down approach is not much more suitable than an exclusively bottom-up approach. Thus a good movie authoring tool must allow the user to work top-down *or* bottom-up as required.

Visualizing the result

Even given the flexibility to work at a desired level of the structure, there remains an overview problem. When completed, the film will have a certain character, and elements of the film which step unnecessarily outside that character may not contribute to the film. An act attempting to convey some particular view will suffer from a scene which is not sufficiently aligned with the purpose of the act.

To design appropriate elements of a film, a filmmaker must have a strong vision of the film. But this vision will change as the film is invented. There is a constant need to stay in touch with the evolving "feel" and character of the film. This becomes quite difficult when working at a low level.

A good movie authoring tool should contain facilities for assisting the user in maintaining an overview of the developing film. One important aspect of MAD is the "play" facility, with which an approximation to the final form of the film can be viewed on the user's workstation at any time. We believe this to be analogous to the way that a word processor user can print the document being written at any time during its development. The "play" facility also allows an author to present the script as a dynamic demonstration of ideas for a film. This is analogous to the use of MIDI files as a machine-playable external form of music, capable of being brought to life more automatically than sheet music.

Other desirable displays or abstractions include a time-line facility, and the accumulation of various statistics pertaining to the script under development.

The System

The current prototype of MAD runs on an Apple Macintosh computer. Apart from the content of the individual items, MAD resembles most of all an outline processor. Although MAD imposes no structure, the top level items will often refer to "acts", the items within acts will refer to "scenes", and so on. Each item itself can have a variety of kinds of data attached to it, and can have subitems. Data can be imported from some standard Macintosh file formats.

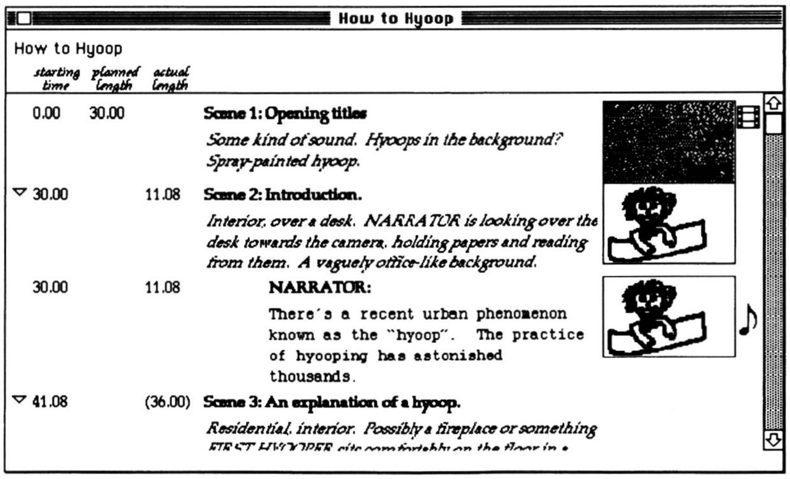
How the structure is manipulated

There are commands to add items into the desired position in the hierarchy. It is also possible to move entire portions of the hierarchy to new places; thus a single operation will take scene 3 and put it before scene 1, along with all of its subitems. Similarly a subitem of scene 3 can be moved to after scene 3 itself, to become a scene 4 rather than a subitem of scene 3.

Since the indentation may get unwieldy with more than a level or two, it is possible to "zoom in" to an item. In Figure 1, the view is the entire movie, whose name appears in the upper-left corner. After zooming in to a given item, that item's title takes the place of that of the film in the upper-left, and only that item's subitems are shown. This means that that item's subitems are moved back over to the left margin rather than being indented unreasonably far to the right.

By zooming in, the user excludes higher levels of detail. It is also possible to exclude *lower* levels of detail with the "expand" and "contract" mechanisms. In the above figure, we see the single shot comprising scene 2. In the left column there is a downward-pointing triangle, with the same meaning as in the Macintosh System 7 Finder "view by name" interface. Pressing this triangle will cause it to turn to the right, and the scene will be contracted, meaning that no subitems are displayed. To see the subitems, the user can zoom in to the scene, or can press the triangle again to expand the item. There are also special "expand all" and "contract all" commands; "contract all" enables the user to view only the current level, with a single operation.

These functions are largely equivalent to those of a typical outline processor.



How to Hyoop

How to Hyoop

starting time	planned length	actual length	
0.00	30.00		**Scene 1: Opening titles** *Some kind of sound. Hyoops in the background? Spray-painted hyoop.*
▽ 30.00		11.08	**Scene 2: Introduction.** *Interior, over a desk. NARRATOR is looking over the desk towards the camera, holding papers and reading from them. A vaguely office-like background.*
30.00		11.08	**NARRATOR:** There's a recent urban phenomenon known as the "hyoop". The practice of hyooping has astonished thousands.
▽ 41.08		(36.00)	**Scene 3: An explanation of a hyoop.** *Residential interior. Possibly a fireplace or something FIRST HYOOPER sits comfortably on the floor in a*

Figure 1: The MAD main view
Scenes, a shot in scene 2, and other data associated with these items in the script for the film "How to Hyoop". The nesting of items is indicated by indentation of the text.

Data represented by the system

In the current system, the main focus is on the written script. Script components are the titles of items (e.g. the names of the scenes), descriptive text, and spoken text (narration or dialogue). The type of text is indicated typographically.

The user can draw storyboard frames, which appear next to the text. Storyboard frames are aligned, so that the column forms a storyboard. Alternatively, a storyboard frame can be imported from the standard Macintosh "PICT" format. This has been used to attach storyboard frames scanned from felt-pen drawings.

Sounds can be recorded using the Macintosh microphone. This feature is intended particularly for trying out narration or dialogue. (Script writers have been known to use a tape-recorder for this purpose.) When sound is attached to an item, an icon of a musical note appears in the rightmost column. This icon can be used to play the sound.

Digitized video can be imported from the standard Macintosh QuickTime format. As with sounds, when a video clip is attached to an item, an icon of a piece of motion picture film appears in the rightmost column. This icon can be used to display the video clip. QuickTime "movies" can contain sound or they can consist solely of video. When they are silent, a sound can be recorded, and even when they are not, their sound can be overridden with a recorded sound. QuickTime movies may also have a "poster frame", which is a user-designated frame somehow typical of the video clip. If the QuickTime movie has a poster frame, this is displayed in the storyboard column, but again, the user may override this with their own storyboard frame.

The system can also represent timing information. Time is represented using an hours, minutes, seconds, and frames representation reminiscent of NTSC time codes. In addition to the starting time (time at which an item begins), each item can have two different times associated with it: a planned length and an actual length. In the simplest case, we only have a plan. For example, the titles sequence is planned to be thirty seconds long, but that's our only idea about it so far. So the planned length would be thirty seconds. In another simple case, we only have some raw data. For example, the interview has been filmed, we know what clip we want to use, and that clip is 31 seconds and 26 frames long. This would be the actual length. A more complex example occurs if we want to trim something down but we're not sure how, in which case the planned length and actual length would both be present, and the planned length would be *shorter* than the actual length.

Actual lengths are entered by the user only for items with no subitems. When an item has subitems, its actual length is automatically calculated by the system as the total of the lengths of the subitems. In the example above, some subitems of scene 3 have planned lengths rather than actual lengths, so the calculated actual length appears in parentheses to indicate its tentative nature. A planned length overrides an actual length in further calculations. Thus, when some trimming is achieved and the actual length value is "good enough", we simply delete the planned length, and the actual length begins to be used by the system for further time calculations.

It is possible to take the "actual length" value from the length of the digitized video clip or sound attached to an item. We have made it easy to override this computerized estimation because we have found that the video or sound available on the computer is frequently only an approximation to the actual images or sounds, which may already exist and thus have precise times associated with them. It is also possible for the user to ask for a simple algorithm to be applied to the spoken text to estimate the time that it would take for an average actor to speak that text.

Playing an approximation to the movie

Successful authoring of a film requires a vision of the final form of the film. As the script begins to take shape, the film should take shape in the author's mind. This vision of the film feeds back into the writing of the script and is crucial if a good script is to result. Again, the importance of visualization is discussed by Katz (1991).

It should be apparent that visualization is difficult, and that any computer assistance would be useful. MAD supports the visualization of the final form of the film via the "play" mechanism, in which a user can play an approximation to the final form of the film at any time, just as how a word processor allows the user to print an approximation to the final form of the document at any time.

It is possible to play the entire movie or to play a single item (which includes playing its subitems). When playing, each item is presented for the correct amount of time. The system shows what it has for each item; it does the best job it can to show the user what that item (scene, shot, whatever) will be like in the final version of the film. It displays the title and any descriptive or spoken text; it displays a storyboard frame if available; it plays a sound if available; it shows a digitized video clip if available.

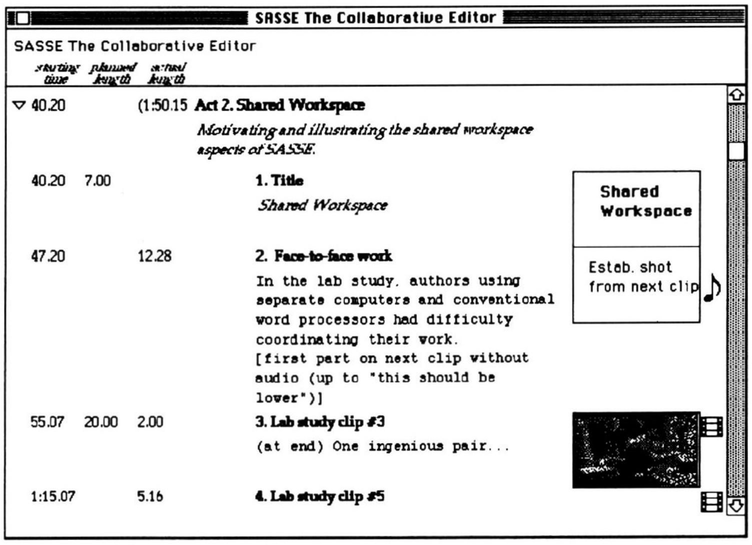
If only text is present, the user has the time to read any spoken text which is displayed, and can attempt to visualize the scene. Even this turns out to be useful because the scene is presented for the correct amount of time and in context before and after other scenes, some of which hopefully have more associated information. When digitized video clips with sound are present and represent the entire contents of their items, MAD becomes a feeble assembly device. The usual use is intermediate between the extremes of text only versus a full set of video clips.

Example Of Use

In September 1993 MAD was used to design a movie entitled "SASSE The Collaborative Editor" for the annual ACM SIGCHI Conference (Baecker, Glass, Mitchell, and Posner, 1994). The movie's purpose was to demonstrate the SASSE collaborative writing system being developed by our research group (Baecker, Nastos, Posner, and Mawby, 1993).

First a very top-level outline of the film was defined, a listing of the acts of the production. The first few acts were then outlined in greater detail in terms of constituent shots. We then took each of the acts in turn and began to draft suitable narration for the script. The narration was recorded so that times could be estimated and judgments made about film flow, timing, and pacing.

The material required three kinds of shots (see Figure 2). Shots of the narrator were indicated in meta-text describing the shot and also in "storyboard frames" containing text only. Longer descriptive meta-text was entered as shooting instructions for the camera crew that would later film user interactions with the SASSE system. Had there been someone on the design team with respectable sketching skills, good storyboard frames would have been created. Finally, where suitable video clips existed, we imported them into MAD and viewed them in the context of the emerging production. Playback of the movie was of course incomplete, but display of the script, reading of the narration, display of the meagre storyboard, and screening of the video clips in correct order and with correct timing sufficed to convey a good sense of the whole and to guide the authoring process.


SASSE The Collaborative Editor

SASSE The Collaborative Editor

starting time	planned length	actual length	
▽ 40.20		(1:50.15	**Act 2. Shared Workspace** *Motivating and illustrating the shared workspace aspects of SASSE.*
40.20	7.00		**1. Title** *Shared Workspace*
47.20		12.28	**2. Face-to-face work** In the lab study, authors using separate computers and conventional word processors had difficulty coordinating their work. [first part on next clip without audio (up to "this should be lower")]
55.07	20.00	2.00	**3. Lab study clip #3** (at end) One ingenious pair...
1:15.07		5.16	**4. Lab study clip #5**

Figure 2: Working on SASSE The Collaborative Editor
Text-only storyboard frame drawings proved useful given our lack of sketching skills. Note the use of bold text in the upper storyboard frame to indicate on-screen titling, as opposed to the second storyboard frame's use of regular text to describe the intended shot.

We used MAD for roughly 6-8 hours in this way. Because the film crew was coming the following day, and MAD at that time was still very clunky and didn't deal with hard copy or still images very well, two members of the team switched to a traditional word processor and markups of paper printouts to produce a final script and shooting instructions for the director and film crew. After filming was completed, a traditional computer-based editing console was used for title generation and post production.

Despite the flaws of the early prototype, MAD allowed us very efficiently to develop and refine a concept for the movie, write and edit the script, revise the script after hearing how it sounded and how it flowed, and preview likely video sequences for inclusion in the film in the context of a playback of a very rough but continually improving approximation to what the final film would be like.

Summary And Conclusions

We have presented a system which uses multimedia to support the authoring of traditional format motion pictures. It allows a user to manipulate structured movie scripts with attached sounds, images, and video clips. The ability to play an approximation to the final film at any time contributes to the author's ability to visualize the final result.

MAD is still under development, as are its underlying ideas. We are using it to attempt to explore the process whereby scripts get created, to determine how best to support this process.

Four approaches to the authoring of motion pictures

More fundamentally, we have identified four types of approaches to the authoring of movies, which we call script-based (text script), storyboard-based, available-shot-based, and flow-based. These are idealized types; in practice some combination of these types of approaches will be used.

The script-based approach corresponds most strongly to traditional practice and is a substantial component of typical introductory filmmaking instruction. In the script-based approach, a text script containing a linear representation of a hierarchical structure is developed using technology such as a word processor that has no specific support for filmmaking. An easy augmentation to this approach is to use an outline processor rather than a word processor.

The storyboard-based approach involves the drawing of storyboard frames in a sequence. A storyboard is often annotated with text such as dialogue or stage directions. This can provide a more visual idea of the film, but only occasionally can it substitute for the text script.

The available-shot-based approach is particularly characteristic of documentaries. When making a documentary, often footage is collected at the same time as the overall flow of the film is designed. Subsequent to all filming, the footage must be examined and organized, and only at this point does the script begin to take shape. The available-shot-based approach is characteristic of some videos prepared for computer conferences, which often take a somewhat documentary form.

Flow-based approaches attempt to keep the entire structure of the film in mind during film authoring and editing. In traditional film editing, the filmmaker may go back and forth across a particular cut many times, adding or removing an amount of film as small as a single frame to attempt to improve the transition. There are also larger flow-based goals; one is the decision of how to divide the total length of the film among its sections. The choices as to the ratio of screen time allotted to various aspects of the film can dramatically affect the result.

It should be clear that a system such as MAD should support all of these approaches. We feel that the current version of our prototype is primarily effective in supporting the script-based and available-shot-based approaches. The storyboard-based approach might appear to be supported, but the current prototype storyboard frame facility fails to provide genuine support for storyboards through a number of functionality and interface problems. Our thoughts about the support of flow-based approaches are not very developed yet, although the planned length versus actual length distinction and the automatic time calculations provide a small measure of support for these kinds of authoring activities. Genuine support for flow-based approaches would involve the ability to display abstractions of film structure or content over time. For example, a "timeline" display (Harrison, Owen, and Baecker, 1994; Owen and Baecker, 1994) in which different rows indicated different speakers could be used to judge whether or not a particular character is speaking too much.

Future research and development directions

It has become clear that different approaches to the authoring of movies require different interfaces. In the current prototype, the interface is not configurable, but a successful system would require a variety of new columns of data, ranging from supplementary text such as actor or prop availability, to alternate "takes" of a video clip, to budget information. With the resulting large number of columns available, it would become crucial to be able to choose which columns were displayed and what fraction of the screen space they took up. Some configurability is also essential to support a true storyboard-based approach; to take one example, the layout as we have it is apparently not compatible with the usual visual layout of a storyboard, because it is too text-script-focused.

Greater direct support for the various datatypes appears to be required. Sometimes one has a collection of video clips without a clear idea of the sequence in which they are to be shown or precisely how they will fit into the text script. Storyboard frames drawn for one shot may be more appropriate for another shot, or a user may wish to begin editing one shot's storyboard frame using another shot's storyboard frame or the scene's storyboard frame as a template. A "slide-sorter" interface for video clips and storyboard frames would permit a user to manipulate these items outside the context of the text script and structure. In some cases, a user will want to import or draw these items before knowing where in the text script they will be placed. It might be possible to build a storyboard-focused interface within the slide-sorter. As well, a slide-sorter interface could ease the task of selecting clips from existing video footage, ideally in concert with the facility in Timelines (Harrison, Owen, and Baecker, 1994) that allows the selection of video clips based on logging information and on displays of film data with respect to a timeline.

To support a flow-based approach, other overview mechanisms are necessary. The "play" mechanism provides one overview mechanism, and the time calculations could be construed as another. It should be possible to construct various static displays resembling timelines, to abstract various attributes as they change over time, as discussed in the previous subsection.

Up to this point, observation of realistic use of the system has been informal only. This has been useful for early feedback but does not comprise a sufficiently sophisticated evaluation procedure. We are currently developing plans to observe and analyze sustained use of MAD.

It is also apparent that once all this data is in the system, there should be a graceful transition to editing. For example, once a video clip is digitized and has been attached to an item in MAD, it should not be necessary to hunt for the video segment on the original tapes to perform the final edit; the time-codes from the videotapes should be tracked by the system as well. In effect, there should be a seamless interface to something such as the Avid Media Composer.

We shall also pursue the connection to multimedia authoring. Although other tools for multimedia authoring exist (e.g. de Mey and Gibbs, 1993; Hamawaka and Rekimoto, 1993; Hardman, van Rossum, and Bulterman, 1993), we believe that in some cases superior presentations can be produced by using an explicitly filmmaking-oriented approach. Although the original goal of MAD is the authoring of traditional films, we intend to generalize the concepts and system to support the authoring of multimedia productions.

Acknowledgments

Russell Owen provided crucial video and QuickTime technical support. Geof Glass and Russell Owen assisted with software development.

Valuable suggestions were contributed by members of the Multimedia Research Group of DGP and by members of The CulTech Collaborative Research Centre at York University.

The DGP laboratory is financially supported in part by NSERC, IRIS, ITRC, CulTech, and Apple Computer.

References

Apple Computer, Inc., 1993. *Inside Macintosh: QuickTime.* Addison-Wesley.

Avid Technology, Inc., 1993. *Media Composer Basic Editing Guide.* Documentation accompanying version 4.5 of the software product. Avid Technology, Inc., Tewksbury, Massachusetts.

Baecker, Ronald M. and Marcus, Aaron, 1990. *Human Factors and Typography for More Readable Programs.* ACM Press, New York.

Baecker, Ronald M., Nastos, Dimitrios, Posner, Ilona R., and Mawby, Kelly L., 1993. The User-Centred Iterative Design of Collaborative Writing Software. *Proceedings of INTERCHI '93.* ACM Press, New York, 399-405, 541.

Baecker, Ronald M., Glass, Geof, Mitchell, Alex, and Posner, Ilona, 1994. SASSE the Collaborative Editor. CHI '94 Video Proceedings, to appear in *SIGGRAPH Video Review.* ACM, New York.

Davenport, Glorianna, Smith, Thomas G. Aguierre, and Pincever, Natalio, 1991. Cinematic Primitives for Multimedia. *Computer Graphics and Applications* 11(4), 67-74.

de Mey, Vicki and Gibbs, Simon. A Multimedia Component Kit. *Proceedings of ACM Multimedia 93* (Anaheim, California, August 1-6, 1993). ACM, New York, 291-300.

Graham, T.C. Nicholas and Cordy, James R., 1990. GVL: A Graphical, Functional Language for the Specification of Output in Programming Languages. *Proceedings of IEEE 1990 International Conference on Computer Languages* (New Orleans, March 1990).

Hamawaka, Rei and Rekimoto, Jun, 1993. Object Composition and Playback Models for Handling Multimedia Data. *Proceedings of ACM Multimedia 93* (Anaheim, California, August 1-6, 1993). ACM, New York, 273-281.

Hardman, Lynda, van Rossum, Guido, and Bulterman, Dick C.A., 1993. Structured Multimedia Authoring. *Proceedings of ACM Multimedia 93* (Anaheim, California, August 1-6, 1993). ACM, New York, 283-289.

Harrison, Beverly L. and Baecker, Ronald M., 1992. Designing Video Annotation and Analysis Systems. *Proceedings of Graphics Interface '92* (Vancouver, May 11-15, 1992). Canadian Information Processing Society, Toronto, 157-166.

Harrison, Beverly L., Owen, Russell, and Baecker, Ronald M., 1994. Timelines: An Interactive System for the Collection and Visualization of Temporal Data. *Proceedings of Graphics Interface '94* (Banff, May 16-20, 1994), this issue.

Katz, Steven D., 1991. *Film Directing Shot by Shot: Visualizing from Concept to Screen.* Michael Wiese Productions, Studio City, California.

Loy, Gareth, 1985. Musicians Make a Standard: The MIDI Phenomenon. *Computer Music Journal* 9(4), 8-26.

MacroMedia, Inc., 1993. *MacroMedia Director version 3.1* (software product with accompanying documentation). MacroMedia, Inc., San Francisco.

Martin, James and McClure, Carma, 1985. *Diagramming Techniques for Analysts and Programmers.* Prentice-Hall, Inc.

Owen, Russell and Baecker, Ronald M., 1994. Timelines: A Tool for the Gathering, Coding, and Analysis of Usability Data. Formal demonstration at ACM CHI '94 (Boston, April 24-28, 1994).

Pea, Roy D., 1991. Learning through Multimedia. *Computer Graphics and Applications* 11(4), 58-66.

Price, Blaine A., Baecker, Ronald M., and Small, Ian S., 1993. A Principled Taxonomy of Software Visualization. *Journal of Visual Languages and Computing* 4, 211-266.

Symantec, 1990. *MORE 3.0 Reference* (documentation accompanying the software product). Symantec Corporation, Cupertino, California.

Woolsey, Kristina Hooper, 1991. Multimedia Scouting. *Computer Graphics and Applications* 11(4), 26-38.