Proceedings of Graphics Interface 2003, June 2003, to appear

Modularity and Hierarchical Structure in the Digital Video Lifecycle

Ron Baecker
University of Toronto
rmb@kmdi.toronto.edu

Eric Smith
Tarkvara Design Inc.
eric@tarkvara.org

Abstract

Despite the multiplicity of data types and rich linking and nesting available in general *multimedia* systems, most *digital video* systems have represented video only as linear sequences of frames and shots. We extend previous work that proposed representing digital video as hierarchically structured documents composed of modular building blocks including outlines, scripts, audio sequences, still images, titles, and motion sequences. We review how such a representation can aid video authoring. We then show how such structure can aid video editing, localizing, browsing, updating, publishing, navigating, and searching. Applications are illustrated with examples from real projects.

Keywords: Multimedia systems, multimedia authoring, digital video, video authoring, video editing.

1 Introduction

Multimedia systems [15] allow the creation and use of interactive computer-based visual presentations that incorporate text, still pictures, animation, audio, and video. Information is presented both sequentially and concurrently. What happens is in part predetermined and in part based on a viewer's actions and reactions.

Motion pictures [16], whether represented on film or as analog or digital video, consist of sequences of still images, or *frames*, which when projected at speeds of 24 or 30 frames per second yield an illusion of motion. Since an hour therefore consists of on the order of 100,000 frames, motion pictures and digital video are typically assembled as sequences of *shots*, where each sequence consists of a sequence of frames.

Our previous research [22,21,2] proposed a method for representing digital video in a richer way than as sequences of sequences of frames, and showed how this structured representation aids authoring.

We begin this paper by reviewing relevant previous work and our new representation. We then show how the representation supports other activities in the life cycle of digital video, with particular focus on video intended for the Internet and the Web. The activities may be clustered into three categories:

- Creation: Authoring, Editing, Localizing
- Production: Browsing, Updating, Publishing
- Access: Navigating, Searching

We conclude by discussing our design and evaluation process and open research problems.

2 Related Work

Systems that support the entry and editing of textual or graphical representations of documents in some application domain are generally known as document editors, authoring tools, or editing tools.

An early multimedia authoring toolkit [19] consisted supported multimedia database access, storyboarding, and editing. An interesting system for multimedia authoring [13,14] employs both a hierarchy view of components and subcomponents and a channel view of what happens over time. An authoring and presentation environment for creating SMIL-compliant documents and a sketching system for creating multimedia storyboards are described in [6] and [3], respectively.

The multimedia authoring system closest in spirit to our work is the Anecdote system [12]. Anecdote allows the use of "surrogate media" to support both top-down design and bottom-up creation, and provides a rich set of complementary representations, the Scene, Link, Timeline, Outline, and Cast Views. An even more ambitious research project [17,18] supports the creation, manipulation, annotation, and archiving/retrieval of media content represented using XML schemas.

There has also been activity in the commercial sector. Desktop video editing tools (pioneered by Avid, http://www.avid.com, and later by Adobe, http://www.adobe.com) allow users to digitize recorded video or digitally record live footage, and then to cut, copy, paste, fade in, fade out, merge, and modify shots. The past decade has seen many similar systems developed, some commercially successful (e.g., http://www.apple.com/finalcutpro), most unsuccessful. Apple and Microsoft also now incorporate movie-editing software as a standard component of their operating systems (i.e., http://www.apple.com/imovie and http://www.microsoft.com/windowsxp/default.asp).

Multimedia authoring tools allow users to create non-interactive or interactive multimedia presentations. In some cases, scripting languages allow users with some programming skills flexibility in tailoring multimedia presentations. An example is Director (see http://www.macromedia.com/software/director/), which employs a theatrical metaphor in a low-level visual scripting environment in which the behaviour of graphical elements known as cast members is arranged with respect to a timeline, or score.

Yet, despite the obvious utility of structure in *multimedia* systems, *digital video* systems from both the academic and commercial sectors have mostly ignored the potential of structured video documents.

3 A System for Structured Video Documents

The system enables design and management of words, images, sounds, and video in digital motion picture preproduction, production, and post-production.

Figure 1: Outline of How to Assemble an Ikea Desk, showing 5 Acts and 9 Scenes within Act 4.

3.1 Key Concepts and Design Goals

Key design goals of what we shall henceforth in this paper call Creator were:

- Hierarchic idea structuring — the ability to represent movies via an outline [Fig. 1], to develop ideas top-down and bottom-up, to modify the structure with ease, and to work at various levels of detail.

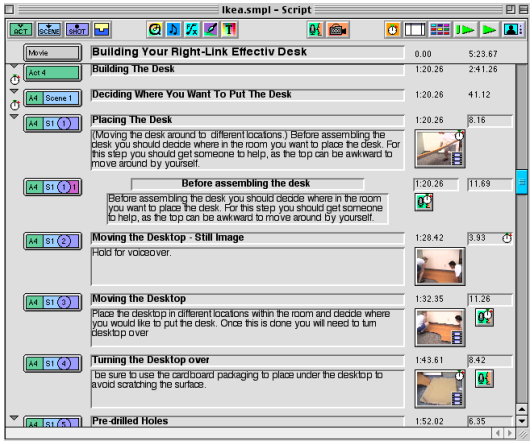

Figure 2: Excerpt from Ikea Script with Act, Scene, Shots, and Spoken. Media elements attached to script and their start times and durations are on the right.

- Multimedia support — integration of outlines [Fig. 1], scripts [Fig. 2], audio (dialogue, narration, music, sound effects), still pictures, storyboards [Fig. 3], titles, and video. All are accessible through appropriate representations and specialized editors, and are linked to the hierarchic movie structure and to a common multimedia database.

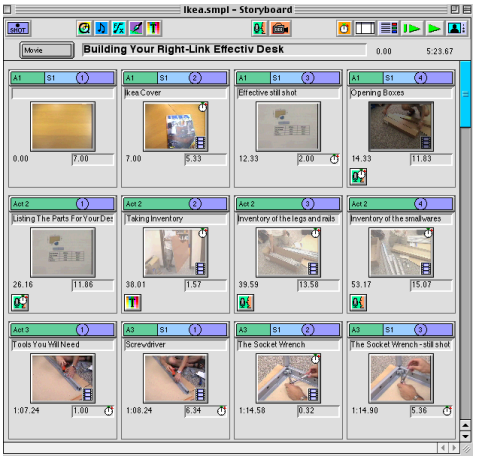

Figure 3: Part of the Ikea Desk Storyboard

- Visualization aids, e.g., getting real-time previews of the movie [Fig. 4] or the best approximation to it anytime in the film development process.

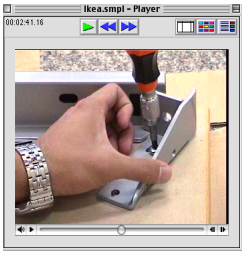

Figure 4: Playback of How to Assemble an Ikea Desk

3.2 Digital Video Representation

The internal representation of a Creator document is a tree. The root of the tree represents the entire movie. Below the root are nodes representing structural elements: Acts, Scenes, Shots, Notes, and Spokens.

A movie typically consists of a number of Acts, each of which consists of several Scenes, each of which is composed of several Shots. Spokens are typically used to encapsulate dialogue that runs over a series of successive shots. Notes are general-purpose elements placed at arbitrary times within the movie. They are used for informational purposes and for fine control over the timing of overlaid graphics, titles, and music.

The leaf nodes of the tree are content elements, such as video clips, audio clips, image files, titles, Bézier-curve drawings, and hyperlinks.

Each structural element has information indicating how its timing is determined. Durations can be explicitly specified, can be based on the total duration of the node's children, or can be based on the duration of a single content element. The flexible timing scheme allows elements to act as building blocks that can be reordered, replaced, and reorganised, while maintaining accurate timing information for the entire movie. A background thread (the "trackifier") ensures that a playable QuickTime preview of the movie is always updated to reflect the current state of the document.

For example, consider a production conceived in terms of three acts. The introduction contains two video shots. The body includes two video sequences, two stills, and another video sequence. The conclusion contains a summary video statement and overlaid credits that begin midway through the summary video.

Spokens and Notes are introduced because music, voice, graphic overlays, and titles do not always associate one-to-one with hierarchy elements, and do not always adhere to the timing of those elements. For example, a music track for Act 1 may be attached directly to the Act 1 node. A Spoken is used to encapsulate a voice-over that extends over Scenes 1 and 2 of Act 2. A Note is used to represent the credits, and its start time is set appropriately within the overall duration of Act 3.

Thus a possible Creator model of this movie is:

Act 1: Introduction (+Note for music)
    Video Shot 1
    Video Shot 2
Act 2: Body
  Act 2 Scene 1 (+Spoken for voice-over)
    Video Shot 1
    Video Shot 2
  Act 2 Scene 2
    Still Image Shot 1
    Still Image Shot 2
  Act 2 Scene 3
    Video Shot 1
Act 3: Conclusion (+Note for titles)
  Act 3 Scene 1
    Video Shot 1

3.3 Implementation

Creator is implemented in Java, with QuickTime as its multimedia engine, and uses the QuickTime for Java library (http://developer.apple.com/quicktime/qtjava/). The current version runs under Mac OS9 and OSX and Windows XP, 2000, NT, 98, and 95.

Creator documents consist of two parts. Structural information is stored in XML files containing serialized versions of the Java objects that represent the documents' structure. Multimedia content files that have been attached to the documents' structure are stored in media folders that are at known locations relative to the XML files. Alternatively, multimedia elements may be stored using an Oracle database, in which case the media folders serve as local caches.

4 Creation

We distinguish three phases in creating a digital video production: *authoring*, to achieve a structured video document; *editing*, to refine and improve the document; and *localizing*, to adapt it for use in different contexts.

4.1 Authoring

Creator's representation of movies as hierarchically structured documents facilitates the planning and organization of a video document in terms of component pieces, much as a book is typically composed out of chapters and these are in turn made up of sections.

Authoring can then proceed in both a top-down and a bottom-up fashion, or in a combination of both. Authoring proceeds top-down when one designs a structure for the video and later fills in the pieces. Authoring proceeds bottom-up when one creates and inputs the pieces and later assembles them into a structure, which is the way many documentary filmmakers work. Filmmakers can move back and forth between top-down design and bottom-up creation, and among tasks such as writing the script, creating voice-overs, inputting music, and working with video sequences.

Figure 1 shows the outline of a movie explaining how to assemble an Ikea desk. Act 1 is an introduction, Act 2 deals with an inventory of the parts, and Act 3 deals with the needed tools. Act 4 constitutes the bulk of the assembly instructions, and is therefore constructed in nine separate scenes. Being able to work with the outline was a thinking aid that helped in the conception and design of the project.

4.2 Editing

Each type of content element has an associated editor. Some editors are simple; others are elaborate, such as the clip editor, which provides a full-featured editor for modifying video and audio clips. Two special "editors" are the audio recorder and the video recorder. These allow users to create new content elements by capturing and modifying audio and video from external sources.

Structural elements are manipulated in the various view windows. Within each of these views, the structural elements can be modified and reorganised as the user sees fit. The Script view provides a hierarchical view of the document organization represented as an outline, as well as the traditional contents of a script such as dialogue and director's notes. In practice it is the view in which most users spend most of their time.

The Storyboard view provides a two-dimensional grid of all shots within a document. This provides a

visual overview, but at the cost of not being able to directly manipulate the hierarchical structure. The Catalog view provides a tabular display with detailed information about content elements associated with the document, and is most useful for "tweaking" content elements later in the production process.

4.3 Localizing

Creator may be used to facilitate the rapid development of structured video presentations that demonstrate software and show users how to accomplish desired tasks. The video demonstrations and explanations can then be integrated into the training, support, or sales sections of a company's Web site, where they can be accessed by users and streamed over the Internet "just in time." In other cases, they are distributed to users via CD/ROM.

An example arises from the increasing demand for high-speed home Internet access. The scarcity of trained installation personnel has encouraged many subscribers to do it themselves. Based on a usability analysis of problems encountered, we produced a multimedia CD showing users what to do [Fig. 5].

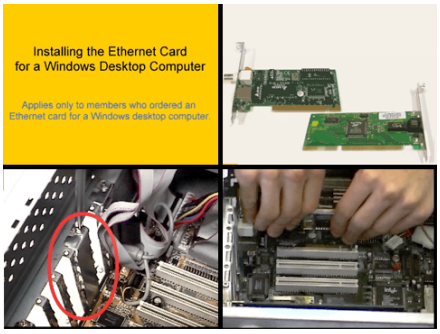

Figure 5: Sample frames from a movie showing how to install an Ethernet card.

Yet we soon began being overwhelmed with the need for large numbers of similar visual explanations:

- Platform variations, e.g., Windows/Mac
- User variations, e.g., English/French, novice/expert
- Market variations, e.g., consumer or business.

The central problem in supporting variants was how to represent dimensions of variation while maintaining the hierarchical structure that had proved so useful. The initial approach was to allow both structural and content elements to have their own variants. However, this proved to be confusing for users, so we decided to restrict variation to structural elements.

One variant is always considered "active". This is important because the program always needs to be able to play back a preview version of the movie.

Each variant has an associated icon (typically a flag for language variants). Users found this insufficient for discriminating variants, so an additional tint was added to the area of the script where variants are displayed [Fig. 6]. Tints and icons may be chosen by users.

The number of dimensions of variation is potentially unlimited, but working with more than two or three dimensions typically overwhelms ordinary users. To help users focus on particular variants, they may hide all variants except the currently active one.

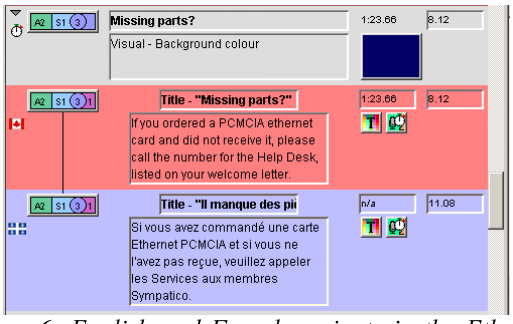

Figure 6: English and French variants in the Ethernet movie. The English version is currently selected.

A typical variant-based document is created with a skeleton consisting of global structural elements that participate in all variants. This skeleton also contains content elements that are global, such as background music or images that do not vary from language to language. The first dimension of variation is usually language, and voice-overs can be attached to language-specific Notes and Spokens. The pace of the movie is typically dictated by voice-overs, so the user chooses these as the source for timing information.

5 Production

We distinguish three aspects of digital video management and production: *browsing*, to allow production personnel to rapidly [re]familiarize themselves with content; *updating*, to keep a video document current; and *publishing*, to transform it into a form that is optimally accessible via the Web and over the Internet, which is one of the most important features of Creator.

5.1 Browsing

Digital video is typically browsed by scanning a storyboard view [Fig. 3] or by playing excerpts from a player view [Fig. 4]. In the latter case, playback may be started and stopped at will.

Because Creator has a rich multiplicity of representations, browsing over the video can involve scanning an outline, the script, the storyboard, the player, and the catalog views [Fig. 7]. One can skim over, or browse, the entire document seeing it as structure (the outline), as text (the script), as still images (the storyboard), as video (the player), and as media files of different types with varying technical specs (the catalog).

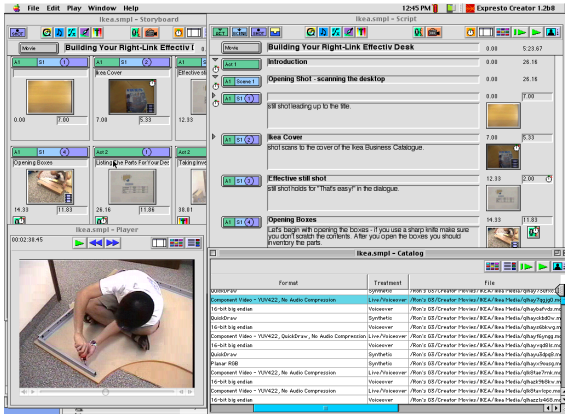

Figure 7. Collage of Creator movie views: storyboard (top left), outline+script (top right), player (bottom left), catalog (bottom right).

Representations can be viewed statically or dynamically. For example, video sequences represented by poster frames in the script and storyboard views "come to life" when clicked within an otherwise static frame. The entire movie or parts of it can be viewed within the collection of representations by starting the player view.

5.2 Updating

Consider the high-speed Internet installation movie. Frequent changes in hardware and software will cause material to soon become obsolete. What is required is an efficient method for updating videos in a small fraction of the time it took for them to be originally created.

The modular building-block approach used for organizing media within a Creator document has proven particularly useful here. Background images, titles, and voice-overs can be replaced while maintaining the overall structure of the movie. Use of the software on real projects led to a number of improvements, notably the addition of extra columns to the catalog view for keeping track of media created outside of the program (e.g., Adobe Photoshop artwork).

Also, the rapid changes required by users raised issues of revision-control and collaborative document sharing. Thus initial experiments were undertaken to explore the feasibility of modifying Creator to store documents using the CVS revision-control system and the Oracle relational data base management system.

5.3 Publishing

In some cases, creators of video documents choose to disseminate them via video cassette or CD/ROM. Yet, increasingly, the Web and the Internet provide a more interesting medium for publishing and dissemination.

Creator's hierarchical structure provides interesting opportunities for web publishing of video documents. In particular, the document can appear on the Web in a way that displays the structure, enabling viewers to access with ease all or part of the document.

This is illustrated by a project for CanadaLeg Inc., developers of the iWalkFree, a simple and elegant walking aid that replaces crutches but can be used hands-free. The result was a 9-minute movie explaining how to assemble, fit, adjust, and use the device.

Publishing the movie this way on the Web [Fig. 8] allows a viewer to see and understand the structure of the production, and to access and view small chunks (scenes), larger chunks (acts), or the entire production. Access and viewing can either be by streaming or by downloading, based on storage-bandwidth tradeoffs.

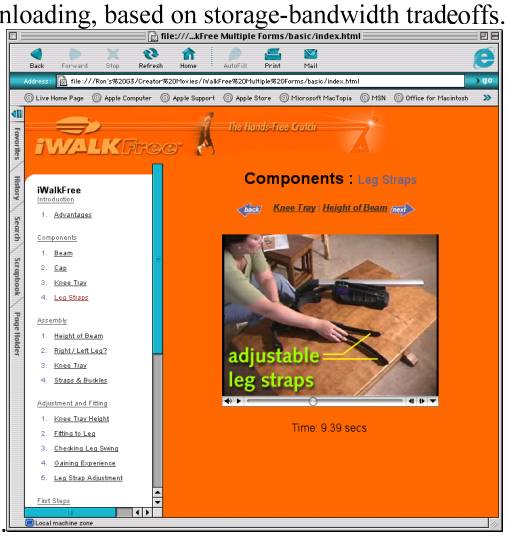

Figure 8: The published iWalkFree movie. The outline is on the left. The movie is paused in the Leg Straps scene of the Components act. Arrows move the viewer to the preceding or following scene.

Most of the hard work of the publishing process is done using XSL (the eXtensible Stylesheet Language, see http://www.w3.org/Style/XSL/). For a particular target, the author creates a set of XSL files that are grouped together. Because the processing is driven by XSL documents, the output can be easily customized to match the look and feel of an existing web-site. Furthermore, XSL is sufficiently powerful that it can be used to generate output in other formats. For example, one template gives us the ability to publish a searchable web-site by generating custom JavaScript code which implements searchable online video documentation without the need for any server-side code. Other templates generate nicely formatted HTML code in a form suitable for printing, and make it possible to publish a Creator document as a set of SMIL files.

In tandem with the XSL processing, Creator includes exporter modules that take the attached multimedia elements and flatten them for final delivery.

Modules have been written for generating output in QuickTime, Real, and Windows Media formats, but the architecture makes it easy to add support for other formats such as MPEG-4 (http://www.m4if.org/mpeg) and MPEG-7 (http://www.mpeg-industry.com).

6 Access

Once production is complete and a video document has been published to the Web, viewers of the document are able to access it. They can view the video in its entirety. They can also *navigate* through the outline representing the hierarchical structure, and *search* for segments dealing with specific words or phrases.

6.1 Navigating

Viewers of instructional or demonstration videos rarely want to see the entire production from start to finish. They would rather scan an outline to get an overview of the whole, and then select particular segments to watch. Creator's publishing module outputs the outline of the production to the Web.

This is illustrated in Figure 8. The viewer can navigate using the outline on the left, and pick a segment to watch. She can then choose another segment using the outline, or move to the preceding or following segment with the left and right arrows above the video frame. Text indicates the current segment, as well as the names of the preceding and following segments.

Since the generation of the HTML pages is entirely driven by XSL templates, there is great flexibility in what navigation interface can be presented to the user. For instance, one template generates a pictorial interface where the user can navigate through the hierarchy using thumbnail images taken from the movie.

6.2 Searching

A final method to facilitate access to a video document is to allow searching for a segment that contains a particular word, phrase, or image. We do not currently support searching by image [25] or by metadata [11], but we support searching by text.

Because a Creator document typically contains the movie's script, searching for words and phrases is straightforward. Automatic or semi-automatic voice track transcription [4] is therefore not needed.

An example appears in Figure 9. Combining the ability to navigate using the outline with the ability to search for any word or phrase significantly aids viewers in finding what they want in a digital video production.

7 Design Process and System Evaluation

We have carried out an iterative, user-centred design process in which concepts and interface design evolved based on observations of users making films with our technology. We did little traditional laboratory usability testing, preferring instead detailed analyses of the work process and products of dozens of filmmakers making many hours of real films.

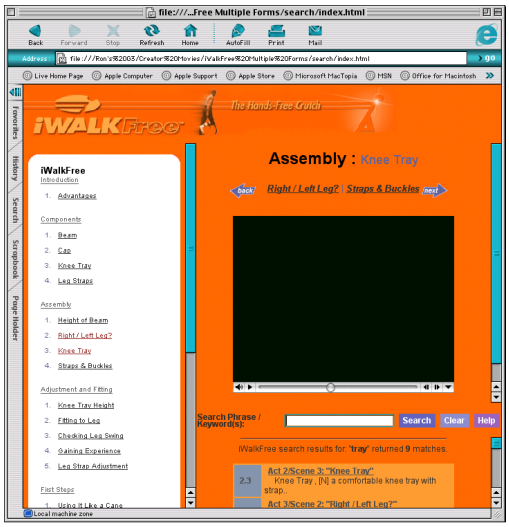

Figure 9: The published iWalkFree movie with search enabled. The search window at the bottom right shows nine places within the movie that "tray" is used..

The process began with a master's thesis [21] implementing a first version in the programming language C. Interviews and observations of the work of early users led to a simplification in the way time was represented, the realization that we needed a storyboard view, titles, full-screen playback, and the ability to record movies onto videotape [2].

We then reimplemented in C++ with much improved functionality and interface. Detailed observations and analysis were made of the use of the new system in two multimedia summer camps [1]. Techniques included background questionnaires, daily questionnaires, interactive feedback sessions, group debriefings, and analysis of movies, artifacts, and activity calendars and audio journals kept by counselors.

We also carried out a controlled within-subjects experiment involving 8 groups of three 7th graders who each made one film with the system and one with a conventional digital editing tool. Results (reported in detail in [20]) included the impacts of technology, experience, and counselor on the complexity and quality of the movies produced and the filmmaking and collaborative work processes employed.

Our third step was to design and implement a "production system" in Java. A general hierarchical structure was specialized to {acts, scenes, shots} based on an analysis of work done with the first two systems. The audio and video clip editors were greatly improved.

Watching what happened when the campers used our software led to a method for the easier and safer management of media resources.

Finally, we began an extensive program of use of this system by staff members, beta testers, and early adopters. Feedback from these activities was sent to the development team and product manager and used to guide decisions on new functionality and interface enhancements. One result, for example, was the development of technologies for web publishing and media management. Another significant achievement was a staff member with little moviemaking experience producing very efficiently an hour of Creator training materials — the world's first video software manual.

Yet, despite following an exemplary iterative design process and what seemed to be a highly user-centred process, the Java production system failed in the marketplace. Besides a variety of reasons associated with management and capitalization of the business, the failure results from an inadequate requirements and user needs analysis. Because we believed that Creator represented a qualitative leap in the ability of filmmakers to conceptualize, design, and create movies, we ignored the evidence that the current generation of filmmakers didn't want new tools. Furthermore, the anticipated new generation was not numerous enough, and was happy to use less interesting but free commercial products bundled into operating systems.

8 Summary and Conclusions

We have presented a novel system for creating, representing, evolving, and using structured digital video documents. Unlike current systems in which digital video is represented as a linear sequence of frames and shots, we model digital video as hierarchically structured documents in which outlines, script, audio, graphics, titles, still images, and motion sequences exist independently. This modular and hierarchic representation assists throughout the digital video lifecycle, including video authoring, editing, localizing, browsing, updating, publishing, navigating, and searching.

We are pleased with the power and flexibility of the representation we have designed and implemented. Authors can work both top-down and bottom-up. The modular representation is supported by a family of media editors that have a consistent interface but are specialized to the needs of particular tasks. The representations also assist flexible browsing, the cost-effective updating of individual media elements in a digital video, and the localizing of videos to deal with different target audiences and families of similar subject matter.

Web publishing and archiving of structured digital video documents is also supported. Users can navigate using the structure of the presentation, and can search on any term or phrase that appears in the script.

9 Future Work

Adjusting timing using the Spokens and the Notes works for very simple presentations, but not as productions become complex. We have designed but not had the resources to implement a timeline view that would be added to the outline/script, storyboard, player, and catalog view. Design of this view involves subtleties OVRe don't exist in typical nonlinear editors because it needs to incorporate the hierarchical structure.

We have planned but not yet implemented the addition of animated sprites to allow videos to incorporate moving overlays and pointers. Although we implemented an experimental screen recorder to produce screen captures for software demonstration and explanation movies, it needs to be reimplemented.

We need better techniques for users to *browse* through the video, looking for interesting aspects. Currently, the family of authoring representations are not available to viewers, so we need new and more powerful representations, for example, video skims [10], visual digests [9], and video summaries [24].

Variants at the moment must be specified at authoring time; different variants must be published as separate video documents. Variants should be selectable at viewing time, allowing viewers to dynamically tailor a movie presentation to their needs. Some recent research [23,5] is relevant to solving this problem.

The software needs to better support group work, to allow individuals to communicate and collaborate with others on large and complex projects. There is a need for interchange mechanisms for importing, exporting, and sharing movies and parts of movies. Users employ a variety of tools to create content, and it should be as easy as possible to move media and structure objects between Creator and other applications. Some form of revision control (see [7,8]) is also required, particularly during the Updating phase.

SMIL 2.0 (http://www.w3.org/TR/smil20/) has evolved to provide a rich set of tags for controlling timing relationships between media objects. We should therefore investigate replacing our internal movie representation with one implemented in SMIL 2.0.

We would like to explore the use of our template architecture for publishing Creator movies in new ways, for example, adding quizzes to visual documentation. Finally, we would like to extend the framework to allow branching and other kinds of nonlinear movies.

Acknowledgements

Creator was originally prototyped at the University of Toronto and then significantly elaborated at Expresto

Software Corp[1]. Current support comes from the Natural Sciences and Engineering Research Council of Canada. Creator was designed and implemented primarily by Eric Smith and Dave Goulden. Agnes Ouellette led Creator client work, assisted by Anne Postic. Ilona Posner led Creator projects in the schools. Mike Ananny, John Hancock, Jonathan Hung, Ali Mazalek, and Kelly Rankin also contributed to this work.

References

1. Baecker, R.M. & Posner, I. (1999). Children as Digital Motion Picture Authors. In Druin, A. (Ed.), *The Design of Children's Technology*, Morgan Kaufmann, 1999, 169-200.
2. Baecker, R.M., Rosenthal, A., Friedlander, N., Smith, E., & Cohen, A. (1996). A Multimedia System for Authoring Motion Pictures, *Proc. ACM Multimedia 96*, 31-42, and in [15], op. cit., 836-847.
3. Bailey, B., Konstan, & Carlis, J.V. (2001). DEMAIS: Designing Multimedia Applications with Interactive Storyboards. *Proc. ACM Multimedia 2001,* 241-250.
4. Brown, M.G., Foote, J.T., Jones, G.J.F., Sparck Jones, K., & Young, S.J. (1996). Open-Vocabulary Speech Indexing for Voice and Video Mail Retrieval. *Proc. ACM Multimedia 96,* 307-316, and in [15], op. cit., 237-246.
5. Bulterman, D.C.A. (1998). User-centered Abstractions for Adaptive Hypermedia Presentations. *Proc. ACM Multimedia 98,* 247-256.
6. Bulterman, D.C.A., Hardman, L., Jansen, J., Mullender, K.S., & Rutledge, L. (1998). GRiNS: A Graphical Interface for Creating and Playing SMIL Documents. *Computer Networks and ISDN Systems* 30 (Sept. 1998), 519-529, and in [15], op. cit., 817-827.
7. Candan, K.S., Prabhakaran, B., & Subrahmanian, V.S. (1996). CHIMP: A Framework for Supporting Distributed Document Authoring and Presentation. *Proc. ACM Multimedia 96,* 329-339.
8. Chiueh, T.-c., Mitra, T., Neogi, A., & Yang, C.-K. (1998). Zodiac: A History-based Interactive Video Authoring System. *Proc. ACM Multimedia 98,* 435-443.
9. Christel, M.G. (1999). Visual Digests for News Video Libraries. *Proc. ACM Multimedia 99,* 303-311.
10. Christel, M.G., Smith, M.A., Taylor, C.R., & Winkler, D.B. (1998). Evolving Video Skims into Useful Multimedia Abstractions. *Proc. ACM CHI 98*, 171-178.
11. Davis, M. (1995). Media Streams: An Iconic Visual Language for Video Representation. In Baecker, R.M., Grudin, J., Buxton, W., & Greenberg, S., *Readings in Human Computer Interaction: Toward the Year 2000,* Morgan Kaufmann, 854-866.
12. Harada, K., Tanaka, E., Ogawa, R., & Hara, Y. (1996). Anecdote: A Multimedia Storyboarding System with Seamless Authoring Support. *Proc. ACM Multimedia 96,* 341-351.
13. Hardman, L., van Rossum, G., & Bulterman, R. (1993). Structured Multimedia Authoring. *Proc. ACM Multimedia 93*, 283-289.
14. Hardman, L. van Rossum, G., Jansen, J., & Mullender, S. (1994). CMIFed: A Transportable Hypermedia Authoring System. *Video Proc. ACM Multimedia 94.*
15. Jaffay, K. & Zhang, H.J. (2002). *Readings in Multimedia Computing and Networking.* Morgan Kaufmann.
16. Katz, S. (1991). *Film Directing Shot by Shot: Visualizing from Concept to Screen.* Michael Wiese Productions, Studio City, CA.
17. Nack. F. & Lindley, C. (2000). Production and Maintenance Environments for Interactive Audiovisual Stories, *Proc. ACM Multimedia Workshop*, Marina Del Rey, CA., 21-24.
18. Nack, F. & Putz, W. (2001). Designing Annotation Before It's Needed, *Proc. ACM Multimedia 2001,* 251-260..
19. Pea, R. (1991). Learning through Multimedia. *Comp. Graphics and Applications* 11(4), 58-66.
20. Posner, I., Baecker, R.M., & Homer, B. (1997). Children Learning Filmmaking with Multimedia Tools, *Proc. Ed-Media'97*, 1997.
21. Rosenthal, A., (1995). Computer Support for Authoring Motion Pictures. M.Sc. Thesis, Department of Computer Science, University of Toronto.
22. Rosenthal, A. & Baecker, R. (1994). Multimedia for Authoring Motion Pictures. *Proc. Graphics Interface '94,* 133-140.
23. Rutledge, L., Ossenbruggen, J.c., Hardman, L., & Bulterman, D.C.A. (1998). A Framework for Generating Adaptable Hypermedia Documents. *Proc. ACM Multimedia 97,* 121-130.
24. Uchihashi, S., Foote, J., Girhensohn, A., & Boreczky, J. (1999). Video *Manga*: Generating Semantically Meaningful Video Summaries. *Proc. ACM Multimedia 99,* 383-392.
25. Zhang, H.J., Low, C.Y., Smoliar, S.W., & Wu, J.H. (1995). Video Parsing, Retrieval, and Browsing: An Integrated and Content-based Solution. *Proc. ACM Multimedia 95*, 15-24, and in [15], op. cit., 350-359.

[1] The technology is now owned by Caseware International Inc., which intends to release it open source. For information, email info@expresto.com