Visual Programming Abstractions for Interactive Multimedia Presentation Authoring

John F. Koegel, John L. Rutledge, Jesse Heines

Interactive Media Group
Department of Computer Science
University of Massachusetts--Lowell
Lowell, MA 01854

Abstract

Various multimedia authoring environments have been developed, resulting in a variety of approaches as to how such presentations are structured and authored. Implicit in these systems is a data model for composing multimedia objects and a user interface model for the authoring activity. Making these models explicit is one goal of this work, since we think it will lead to the design of easier-to-use and more powerful tools. We distinguish the structure of the multimedia objects being composed and the different presentation states that the viewer experiences. The latter is what the author is most concerned with; the former is what most tools allow the author to directly manipulate. We illustrate these ideas with examples of multimedia authoring tools that we have developed and suggest several directions for improvement.

1.0 Introduction

The development of tools which make the creation and composition of multimedia elements easy to perform is a prerequisite for widespread use of multimedia in information systems. Currently available tools [1,3,5] show some of the variety and power of multimedia authoring but on the whole a systematic model of such tools has not appeared. Multimedia authoring is typically viewed as the problem of composing media forms into spatial, temporal, and interactive associations to create an interactive multimedia presentation. The use of multimedia authoring includes not only interactive presentation applications but also:

- Time-based annotation
- Compound multimedia documents
- Multimedia slide shows
- Computer-based training
- Hypermedia environments
- Virtual reality
- Collaborative multimedia communication

While there is clearly overlap between these different areas, each can be viewed as having a different emphasis. We anticipate that a variety of multimedia authoring models will be developed that will be specialized to the above applications as well as others. The user will then have the choice of selecting a tool that is best fitted to the type of composition being performed.

For several years we have been developing different authoring tools that can be used to create multimedia compositions for a number of the applications listed above. We have been particularly interested in addressing certain aspects of multimedia authoring, including:

1. Authoring modes which are distinct from user modes
2. Multimedia composition, in particular temporal composition, with the assumption that monomedia editors either exist or will be developed
3. User interface paradigms which take advantage of the visual element, such as visual programming
4. Identifying abstractions for use in designing user interfaces for such tools as well as for designing the internal representation of the multimedia information

In this paper we describe two multimedia authoring tools that we have developed that use visual programming techniques. We use these examples as vehicles for discussion of the underlying data model and user interface model.

2.0 Composition Models

2.1 Data Models

The *data model* represents the internal organization of the media objects and their interrelationships that is being manipulated indirectly by the user to create the presentation. A model for multimedia composition must address spatial, temporal, and interactive dimensions. Contemporary systems for the most part do not make their data models explicit although the model manipulated by the user might directly correspond to the internal representation. HyTime [6] defines a logical model and MHEG [9] defines an encoding model for multimedia compositions.

2.2 User Interface Models

The *user interface model* is the set of screen objects and relationships that the author works with to create and edit the presentation, and the associated interaction steps. Two common paradigms are script-based editors and iconic visual programming tools in which the author combines the desired media using a programming model. The media may be edited in the same tool or using other applications such as image editors, sound editors, etc. It is desirable that the user interface model correspond closely to the authoring model, i.e., the conceptual and perceptual objects and relationships that the author is concerned with.

3.0 Two Iconic Multimedia Authoring Tools

3.1 Eyes M/M

Eyes [4] is an iconic visual programming environment designed to be used with many different applications. Eyes provides a general-purpose iconic graph editor which can be customized using application-specific icon class definitions. Class definition files describe the icons, their data fields, and the possible interconnections between different type of icons. The graph editor is used to create a graph of interconnected objects which is then sent to one or more backend application processes which interpret the graph and execute it.

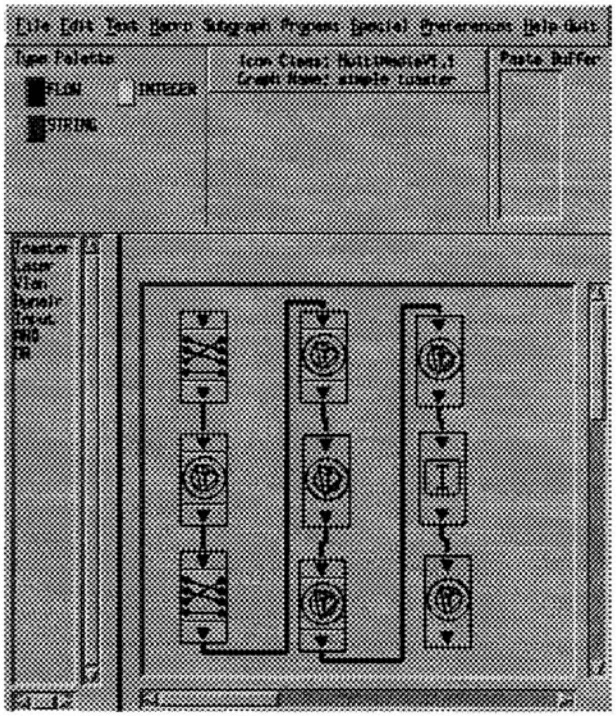
A multimedia authoring system, Eyes M/M, has been developed using Eyes as the user interface (Figure 1). The icons are combined in a graph which is equivalent to a timed Petri net (TPN) [8]. The graph represents the time flow of the presentation. Branches in the graph represent parallel execution paths. Joins in the graph represent synchronization points of two or more parallel paths. Groups of icons can be associated as macros for re-use.

There are three categories of icons: action icons, input icons, and flow icons. Action icons are the presentation elements, such as showing a video segment or displaying an image. Input icons provide user input. Flow icons are used to provide synchronization. Action icons provide atomic synchronization in two ways: 1) an optional delay which precedes the action; 2) an asynchronous mode in which graph execution does not block while an action is being performed.

Two flow control icons have been implemented. They are the AND and OR icon. This provide a means for combining parallel time flows. The OR icon waits until a flow reaches one of its two inputs and then activates the output flow. When the second input flow reaches the icon it is ignored. The AND icon waits until all input flows have reached it before activating the output flow.

An example of a simple sequential presentation is shown in Figure 1. The icons corresponding to operations involving a video switch, a video-effects unit, and input from the user. More elaborate graphs involving a range of multimedia devices and parallel paths have been constructed using Eyes M/M.

FIGURE 1. Eyes M/M


File Edit Text Macro Subgraph Program Special Preferences Help Quit
Type Palette
FLOW
INTEGER
STRING
Graph Name: simple toaster
Paste Buffer
Toaster
Laser
Input
AND
OR

3.2 MediaVisualizer

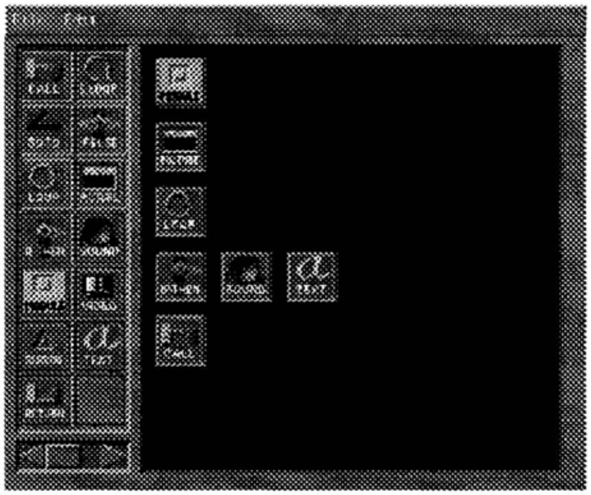
MediaVisualizer (Figure 2) is a prototype authoring tool which uses a procedural model for composing presentations. The editor consist of a two-dimensional grid for placing icons with icons for each type of media and several standard control icons. The horizontal dimension of the graph is used to show nesting of procedural blocks. The vertical dimension is used to show the sequence of execution. The example in Figure 2 shows a loop which will display a picture and then perform a conditional test. The test determines whether an audio or video object is presented.

4.0 Evaluation

Eyes M/M and MediaVisualizer represent two iconic approaches to creating time-based multimedia compositions. One uses a time-representation based on TPNs while the other uses a functional model (see [7]). As a user interface model, the TPN representation makes parallel presentation steps more apparent.

While these representations provide flexible tools for representing and editing time-based multimedia compositions, they are more appropriate as internal data models than as user interface paradigms. In particular, these models focus on the *programming* of the presentation which is a low-level view of the authoring activity.

FIGURE 2. MedlaVlsuallzer



The authoring process typically starts with a storyboard which lays out the general organization and content of the presentation. The storyboard evolves as the media are collected and organized; new ideas and refinements to the presentation are added as the presentation takes shape. The author/artist replays parts of the presentation during this refinement process.

Current tools could be improved by providing an interface model which fits this authoring process. There are three conceptual levels in which an author works:

1. Storyboard
2. Media selection and layout
3. User input and flow of control

Each of these is parallel in time and could be represented as a parallel track in which time is in the horizontal direction. The storyboard track would permit freehand drawing of a series of panels. The media layout track shows the corresponding presentation sections with pictures, video windows, text, graphics, and interaction areas. Typically a storyboard panel would expand into a number of presentation panels. The control track would use a visual programming graph to define the input and link associations for the media layout track.

Most iconic authoring tools take limited advantage of the visual medium which they control. The icons represent data types rather than a miniature view of the presentation itself. Consequently the author has to shift between play and edit modes during editing. Two ways to provide the author with the visual context during icon manipulation are: 1) change the icons to show the (scaled down) visual itself, and 2) show a miniature the presentation on a timeline in a separate area of the screen.

5.0 Conclusions and Further Directions

Visual programming techniques will be an important component in the development of multimedia authoring tools. Multimedia authoring adds several new considerations to visual programming including temporal composition and providing visual context to the author. We have developed two tools which illustrate different temporal composition mechanisms and have suggested several methods for providing visual context. Several directions for further work are evident, including formalizing the temporal composition models and experimenting with more visually oriented authoring interfaces.

6.0 References

[1] Authorware, Inc. Authorware Professional Manual. Redwood City, CA, 1989.

[2] Chang, S.-K. Visual Languages: A Tutorial and a Survey. IEEE Software. Jan. 1987. pp. 29-39.

[3] Commodore Electronics, Ltd., AmigaVision Authoring System Manual. West Chester, PA, 1990.

[4] Eyes 2.0 Reference Manual. Center for Productivity Enhancement. University of Massachusetts--Lowell, 1992.

[5] AimTech Corp. IconAuthor User Manual. Nashua, NH, 1991.

[6] ISO JTC 1/SC 18/WG 8 DIS 10744 Hypermedia/Time-Based Structuring Language (HyTime), Oct. 1991.

[7] Koegel. J., Keskin, C., and Rutledge, J. Toolkits for Multimedia Interface Design. Xhibition 92. pp. 275-285.

[8] Little, T., and Ghafoor, A. Network Considerations for Distributed Multimedia Object Composition and Communication. IEEE Network Magazine. Nov. 1990, pp. 32 - 49.

[9] ISO/IEC JTC1/SC29/WG12. MHEG Working Document "S.6", June 1992.