DEMAIS: Designing Multimedia Applications with Interactive Storyboards

Brian P. Bailey, Joseph A. Konstan, and John V. Carlis
University of Minnesota
Department of Computer Science and Engineering
Minneapolis, MN 55455
{bailey, konstan, carlis}@cs.umn.edu

ABSTRACT

To create an innovative interactive multimedia application, a multimedia designer needs to rapidly explore numerous behavioral design ideas early in the design process, as creating innovative behavior is the cornerstone of creating innovative multimedia. Current tools and techniques do not support a designer's need for early behavior exploration, limiting her ability to rapidly explore and effectively communicate behavioral design ideas. To address this need, we have developed a sketch-based, interactive multimedia storyboard tool that uses a designer's ink strokes and textual annotations as an input design vocabulary. By operationalizing this vocabulary, the tool transforms an otherwise static sketch into a working example. The behavioral sketch can be quickly edited using gestures and an expressive visual language. By enabling a designer to explore and communicate behavioral design ideas using working examples early in the design process, our tool facilitates the creation of a more effective, compelling, and entertaining multimedia application.

Keywords

Authoring, Gestures, Multimedia Design, Storyboards

1. INTRODUCTION

To create an innovative interactive multimedia application, a designer must rapidly explore, compare, and communicate numerous behavioral design ideas early in the design process. Unlike hypermedia, web, and user interface design, multimedia design focuses on creating a unique user experience through the coherent use of attractive content, engaging interaction, and time-based information display. In this paper, we use the term behavior to collectively refer to the design space dimensions of user interaction and time-based information display.

To gain a better understanding of the creative multimedia design process and the challenges a designer faces when confronted with a new project, we surveyed (13) and interviewed (12) professional multimedia designers, primarily located in the Minneapolis area. Most of the designers had at least 5 years of experience building interactive CD-ROMs and web sites, producing television commercials, or developing instructional training applications. In addition, the first author spent 6 months as part of a development team responsible for building commercial, interactive CD-ROMs.

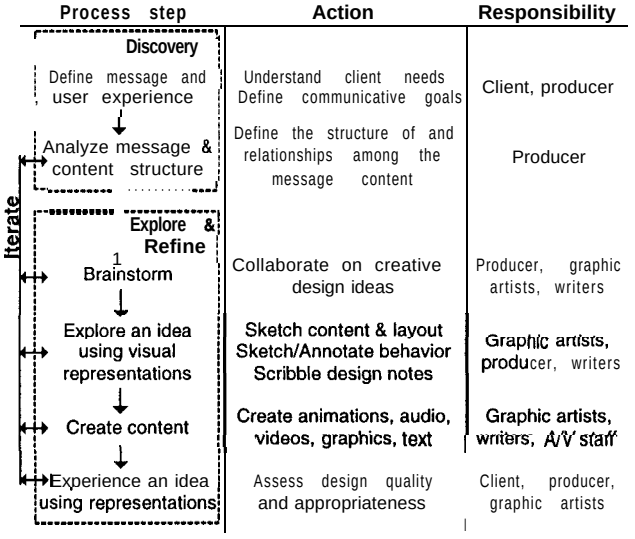

Figure 1. The early multimedia design process. Although the process steps are itemized in a linear fashion, they are often **followed in a non-linear and** ad-hoc sequence. A design team iterates among the steps until an acceptable design emerges.

The surveys were conducted to gather information on the multimedia design process, design artifacts produced within that process, and the tools used to produce those artifacts. The face-to-face interviews were conducted to gain an in-depth understanding of the design process in the context of an ongoing or recently completed project. The interviews were especially informative as we were able to ask follow-up questions and hear anecdotal design stories that could not otherwise be communicated through a structured survey. Figure 1 provides a macro level view of the early multimedia design process.

Although we learned many lessons from the surveys, interviews, and direct experience, the most important lessons of multimedia design is that a designer:

- *Resists using any formal notation, authoring tool or other prototyping application early in the design process.* Designers felt that current authoring tools such as Director [21] and Authorware [25] require too much early investment in terms of time and money. If a large initial investment is made, then time and budget constraints minimize the potential for future

MM'01, Sept. 30-Oct. 5.2001, Ottawa, Canada.

changes. Designers also stated that these tools are overly focused on building a final-form application and best suited for implementing a final design, not for helping them determine what that final design should be.

- *Relies heavily on paper storyboards to capture both the behavioral and non-behavioral dimensions of the design space.* Even with access to high-quality production tools, a designer still prefers pencil and paper early in the design process. Using a pencil, a designer sketches visual layout and interaction, writes behavioral annotations, and scribbles other design notes on a storyboard. For example, a designer might annotate or sketch the behavior, *"When a user clicks on this text, change image X to image Y."*
- *Struggles to explore, compare, and communicate behavioral design ideas early in the design* process. From our experiences of talking with designers, hearing design stories, and inspecting numerous storyboards loaned to us, a designer unmistakably struggles to visualize the behavior of an early design idea. Yet designers agree that creating innovative behavior is the cornerstone of creating innovative multimedia.

A design story will help illustrate this last design lesson. A designer was challenged with building an innovative navigation concept for a local artist's work. The designer communicated at length with the artist, discovering that the artist had two primary themes in his work that might be leveraged in a creative navigation concept. Her idea was to create two rows of concept keywords and enable the user to independently slide each row of keywords across the other. As the keywords aligned, a menu would appear enabling the user to navigate to the artist's work depicting the overlap of those two concepts. However, the designer discussed at length her struggle to communicate that idea to both her colleagues and client as they had difficulty visualizing how that design would *feel* for an end-user. Although prototyping the idea with an authoring tool was discussed, the investment was too large, especially for an idea that might be scrapped in favor of another. In the end, the designer used onion skin paper to physically demonstrate how a user could drag each set of concept keywords across the other as well as the menu display that would appear. If you had difficulty visualizing this interaction, then you understand the struggle a designer routinely faces.

To summarize, a designer needs to rapidly explore and effectively communicate numerous behavioral design ideas early in the design process, but struggles to do so. In addition, research has convincingly shown that the effective use of the behavioral dimensions in a multimedia application helps a user to better absorb [3, 9, 23] and longer retain information [16,241, furthering the research need to address this struggle.

To address a designer's struggle with early behavior exploration, we have developed a novel design tool called DEMAIS (DEsigning Multimedia Applications with Interactive Storyboards). DEMAIS is a sketch-based, interactive multimedia storyboard tool that uses a designer's ink strokes and textual annotations as an input design vocabulary. By operationalizing this vocabulary, the tool transforms an otherwise *static* sketch into a *working* example, facilitating *experience-based* exploration (exploring an idea through a working example). The behavioral sketch can be quickly edited using gestures and an expressive visual language. By enabling a designer to explore and communicate behavioral design ideas using working examples early in the design process, our tool facilitates the creation of a more effective, compelling, and entertaining multimedia application.

We summarize our motivation and provide a context for later discussion with an example. Imagine that you are asked to create an interactive CD-ROM regarding the early, and now famous, U.S. expedition led by Lewis and Clark in the early 1800s [18]. The goal of your application is to create a unique user experience enabling a user to learn about and explore the:

- Biographical information of the expedition members
- Goals of and historical context surrounding the expedition
- Path traveled through the U.S. states and territories
- Original markers that still exist today depicting the path
- Physical items used to overcome natural obstacles
- Unusual sites and sounds encountered along the way

This is only a partial list of information that one might include in the application. Our goal is not to define an exhaustive set of requirements for such an application, but to convey that this is a canonical design example that a multimedia designer often faces. An effective multimedia design tool must support a designer in her exploration of the design space for such an application.

The rest of this paper is organized as follows. In Section 2, we discuss design tools used in other domains and how our design tool differs from them. In Section 3, we describe the design components of DEMAIS and how a designer uses those components to design an interactive multimedia application. The implementation of DEMAIS as well as some limitations are discussed in Section 4. In Section 5, we report on an informal evaluation of DEMAIS with two experienced designers and outline directions for future work. In Section 6, we summarize the main contributions of this work.

2. RELATED WORK

In this section, we discuss several design tools used in a variety of domains, focusing on how these tools fall short of meeting the unique needs of a multimedia designer and how our design tool differs from them.

2.1 Tools for Sketching Early Design Ideas

Regardless of the domain, a designer almost always sketches her initial design ideas on paper rather than using a computerized design tool [1 1]. A paper sketch can be made quickly (unimpeded), is portable, and allows a designer to be ambiguous, imprecise, and vague. On the other hand, a paper sketch does not support modeling, simulation, or automated critiquing of the sketch, all of which a computerized tool can support. To combine the advantages of paper with the advantages of a computerized tool, an electronic sketching tool interprets a designer's ink strokes in order to build a semantic representation of the sketch. Using this semantic representation as input, a variety of rules and computation can then be applied. Electronic sketching tools have been built for user interface [15, 17] and web site [19] design.

SILK [17] is an electronic sketching tool for early user interface design. Using SILK, a user interface designer sketches a series of storyboards and illustrates transitions among them. By operationalizing the illustrated transitions, SILK enables a designer to experience those transitions beyond that provided by

paper mockups alone. SILK also attempts to recognize sketched widgets in order to provide default behaviors for them.

DENIM [19] is an electronic sketching tool for early web interface design. DENIM provides a zoomable canvas enabling a web designer to quickly move among the sitemap, storyboard, and sketch views. Similar to using SILK, a designer can sketch a navigational link from a source widget to a destination page. When requested, DENIM operationalizes the navigational links enabling a designer to experience navigating among the information content.

Although user interface and web site design are related to multimedia design, the design tools used within these domains do not meet the needs of a multimedia designer. In addition to structuring pages containing text and images and experiencing content navigation, a multimedia designer must also be able to explore creative temporal layouts, more expressive interaction, and the use of dynamic media such as audio, video, and animation within an application.

2.2 Tools for Multimedia Design

To offer design advice to a multimedia designer, Faraday [8] built a design tool with an embedded critic for reporting attentional focus conflicts within a multimedia application. Because the tool was designed specifically for attentional focus, a designer cannot use it to explore other dimensions of the multimedia design space such as spatial layout or interaction. However, an attentional focus critic may be a valuable addition to our design tool.

Anecdote [12], and its successor MediaDesc [7], are multimedia design tools providing some support for the early exploration of a design idea. These tools support the early design process by enabling a designer to create a collection of annotated storyboards and then connect them with navigational links or sequence them in time. The tools also support several views of a design including an outline, timeline, scene, link, cast, and play view. Through these synchronized views, a designer can explore design alternatives at different levels of abstraction.

Although Anecdote and MediaDesc provide a reasonable start for a multimedia design tool, our design tool extends and strengthens many of the concepts found in them. First, our design tool operationalizes a storyboard using both the ink strokes and textual annotations made by a designer, neither of which are operationalized by Anecdote or MediaDesc. Second, our design tool incorporates an expressive visual language enabling a designer to rapidly explore alternative behavioral design ideas and serves as a communication medium between the tool and a designer and between two or more designers.

2.3 Tools for Multimedia Authoring

A multimedia authoring/prototyping tool facilitates the implementation of an application by providing abstractions for visual and temporal layout, media playback, and user interaction. Examples of authoring tools include HyperCard [1], Director [21], Authorware [25], and Grins [6]. Although an authoring tool offers a designer expressive power and flexibility, the associated programming language or metaphor is often too difficult to learn and use, overly focused on building a final form application, and constraining in that it forces a designer to design within that language or metaphor. These limitations were verified in the surveys and interviews we conducted. Furthermore, the reported limitations of using an authoring tool early in the multimedia design process are consistent with the reported limitations of using a user interface prototyping tool such as Visual Basic early in the user interface design process [15, 17,261.

Our tool is a design tool, not a *prototyping* tool. Our tool is targeted for the early stages of multimedia design, when a designer's ideas are still rough and evolving. Our tool enables a designer to rapidly explore the behavioral dimensions of the design space early in the design process. Once an acceptable design emerges, then a prototype of that design can be built, if desired.

2.4 Using a Synchronization Model as a Design Tool

A synchronization model is a formal technique for specifying the temporal relationships among media objects within a multimedia application as well as the effects of user interaction on those relationships. Although a synchronization model is appropriate for the specification of a multimedia application, it is inappropriate for the ***design*** of a multimedia application. Synchronization models provide inadequate support for modeling innovative user interaction [5], which is required for an innovative multimedia application. Also, in order for a specification to be executable, it must be both precise and complete. However, early in the design process, a designer needs to be ambiguous, imprecise, and vague [11]. From our previous research in multimedia synchronization [5], we learned that solving the problem of synchronization specification is different from solving the problem of more effective application design.

2.5 Tools for Video and Film Design

Similar to our observations in the multimedia domain, Mackay and Pagani [20] recognized that a video designer routinely uses paper storyboards early in the video production process even with the availability of high-quality video editing tools. As a result, Mackay and Pagani created a video design tool called Video Mosaic. Video Mosaic provided an augmented reality environment containing both a physical and electronic version of a paper storyboard. The idea was that a designer could manipulate a physical storyboard and then experience the modifications through the corresponding electronic version.

The MAD (Movie Authoring and Design) [2] tool facilitated the process of designing a linear visual presentation such as a film or lecture presentation, The primary means of facilitation was a "play" mode enabling a designer to experience the presentation story structure as it existed hitherto. Empirical evidence gathered by the authors indicated that the play mode was one of the most beneficial aspects of the design tool. MAD is an excellent example of how a software tool can aid a designer in exploring, comparing, and communicating early design ideas leading to a more effective presentation.

Although both Video Mosaic and MAD have proven useful within their respective domains, they do not support the exploration of user interaction or time-based information display required for an interactive multimedia application, which is the focus of our tool.

3. DEMAIS – COMPONENTS AND USE

DEMAIS provides a designer with a behavioral sandbox in which to quickly sketch alternative behavioral ideas, analogous to how a designer uses pencil and paper to sketch alternative visual layouts. DEMAIS enables a designer to rapidly explore, compare, and communicate alternative behavioral design ideas using working examples as opposed to using static sketches, scribbled notes, and imagination. Specifically, DEMAIS enables a designer to:

- Sketch and annotate a set of storyboards using a stylus and electronic canvas just es she would using pencil and paper
- Edit a storyboard using gestures
- Develop voice narrations and import images, audio, and video into a storyboard with a tap of the stylus
- Sketch synchronization among the voice narrations, audio, video, and other sketched content
- Define interactive and time-based behavior using ink strokes and textual annotations
- Edit behavior using an expressive visual language
- Operationalize the behavior sketched and annotated on a set of storyboards to obtain a working example

Our goal is to help a designer create a more innovative interactive multimedia application, such as for the Lewis and Clark experience discussed earlier, by enabling him to directly experience behavioral design ideas early in the design process.

For the remainder of this section, we describe the design components of DEMAIS and how a designer uses those components to design an interactive multimedia application. First, we discuss how a designer designs content using the storyboard and narration editors and manages that content using the content manager. Second, we describe how a designer explores behavior using visual language icons, behavioral ink strokes, annotations, and the multi-view editor. Last, we describe how a designer operationalizes a behavioral sketch to obtain a working example.

3.1 Designing Content: The Storyboard and Narration Editors

In DEMAIS, a designer designs content using the storyboard and narration editors. The multi-view editor enables a designer to simultaneously view multiple storyboards and narrations and sketch behavior among them. Because the multi-view editor is used to sketch behavior rather then content, we save further discussion of it until Section 3.2.4. Next, we discuss the storyboard and narration editors es well es the content manager.

3.1.1 Storyboard Editor

The storyboard editor is an electronic canvas on which a designer can freely sketch and scribble notes using a stylus. A behavioral sketch created with the storyboard editor showing a possible scenario for part of the Lewis and Clark experience is shown in Figure 2. The storyboard editor supports cut, copy, paste, and selection gestures similar to those in DENIM.

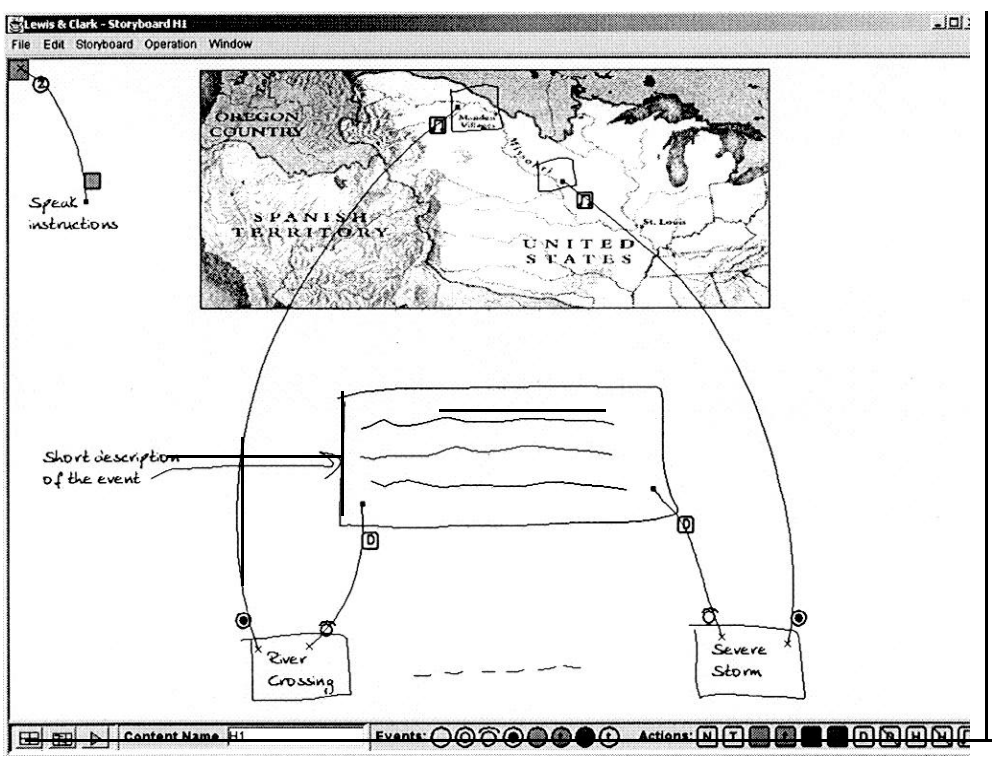

Figure 2. The DEMAIS storyboard editor showing a behavioral sketch for the Lewis and Clark interactive experience. With the bottom two rectangles, a designer is exploring the idea of having the user drag and drop en icon representing a hardship encountered by Lewis and Clark to the corresponding geographic location et which it occurred. If correct, the user hears a narration telling the story further. With those same rectangles. the designer is also exploring the idea of displaying a description of the represented event whenever the user rolls the mouse over them. To aid the user, the designer is thinking that a few seconds after navigating to this screen. the user should hear a short set of narrated instructions. To test and communicate the behavioral ideas sketched on this storyboard. the designer presses the 'play" button in the lower left corner, operationalizing the sketched behavior. This behavioral sketch took just a few seconds to create.

A storyboard sketch consists of ink strokes, recognized objects, plain text objects, behavioral ink strokes and visual language icons, and annotations. An ink stroke is a stroke defined from the point at which a designer presses the stylus to the canvas to the point at which the stylus is lifted. After an ink stroke is created, the system attempts to recognize it as an object or behavior.

Recognizing an ink stroke as an object is performed using a modified implementation of Rubine's gesture classifier [13]. Currently, the classifier has only been trained to recognize an ink stroke as a rectangular ink stroke (rectangle), although additional object recognition may be added in the future. Recognizing an ink stroke as rectangular is performed by comparing that ink stroke to a set of pre-defined rectangle examples along several dimensions, including the distance between the start and end point, initial sine and cosine, and sharpness of the ink stroke [13]. If the weighted inter-dimensional distance between the ink stroke and rectangle examples is "small," then the system marks that ink stroke as a rectangular ink stroke.

Within a rectangular ink stroke, a designer taps the stylus once to bring up a tile browser allowing her to select an image, audio, or video tile. Once selected, the image, audio (its controller), or video is positioned at and scaled to the dimensions of the rectangular ink stroke. The rectangle is then marked as an image, audio, or video rectangle. This sequence of steps was followed to import the image of the U.S. map shown in Figure 2.

If an ink stroke is not recognized as a rectangular ink stroke, then the system attempts to recognize it as a behavioral ink stroke, which is discussed in Section 3.2.2. If the system does not recognize the ink stroke as a rectangular or behavioral ink stroke, then it is displayed and stored, hut has no current interpreted meaning to the system.

The storyboard editor also allows a designer to enter any amount of text onto the canvas. To obtain a text entry cursor, a designer presses and holds the stylus to the canvas for a short moment. Afterward, a blinking cursor appears and a designer enters the desired text using the keyboard, The text is displayed using a handwriting font in order to keep a sketchy feel to the storyboard content. After the text has been entered, DEMAIS attempts to parse that text as an annotation (NSync role). If the text can he parsed without error, then it is marked as an annotation and passed to the runtime engine whenever the design is operationalized. If the text cannot be parsed without error, then it is displayed and stored, but has no interpreted meaning to the system. Annotations are discussed further in Section 3.2.3.

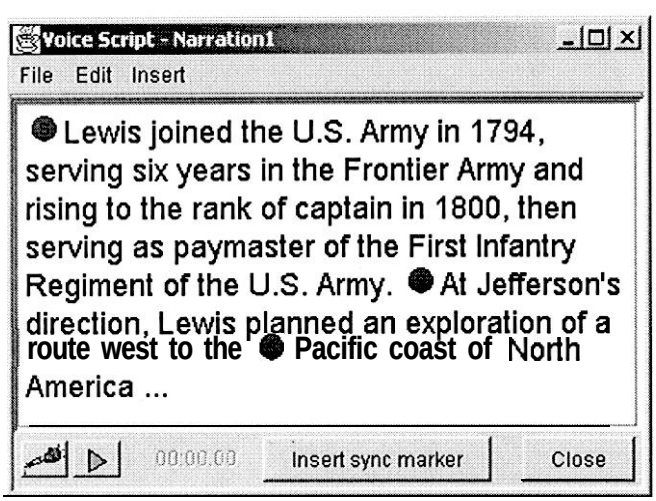

Figure 3. The narration (voice script) editor. A designer can hear the text spoken aloud, record his own voice, and insert synchronization markers at appropriate points in the text.

The storyboard editor displays a start symbol in the upper left corner representing time 0 for a storyboard. The start symbol provides an anchor from which a designer can draw a behavioral ink stroke indicating that an action be performed some time after the user navigates to that storyboard. Figure 7 shows an example of using this symbol to begin playback of a video a few seconds after a user navigates to the containing storyboard.

3.1.2 Narration Editor

Many of the designers we surveyed and interviewed considered the design of voice narrations to he an integral part of the early multimedia design process. Thus, DEMAIS allows a designer to build one or more voice narrations using the narration editor shown in Figure 3. To gain a rough feel for how the narration sounds as it is being developed, a designer can either instruct the system to read the text aloud using a text-to-speech synthesizer or record his own voice as he reads the text aloud himself,

Using the narration editor, a designer can insert any number of synchronization markers into the narration text. A synchronization marker enables a designer to specify that an action he performed when a specific point in the narration is reached during playback. For example, a designer may want to highlight a geographic region on a map whenever that region is referenced in the narration. The value of exploring synchronization markers early in the design process is to:

- ***Achieve a desirable visual*** **pace.** Visual transitions synchronized to a voice narration should not occur too quickly, nor should they occur too slowly. Achieving the desired pace may also help a designer determine the length of the narration as well as the amount of visual content that needs to he developed.
- ***Properly synchronize*** content. A strength of teaching with multimedia is the ability to reinforce a message using properly synchronized content [3, 9, 241. If the content is not properly synchronized, the message may not be conveyed.

To assign an action to a narration's synchronization marker, a designer drags the appropriate narration from the content manager and drops it in the multi-view editor. See Figures 4 and 8. In the

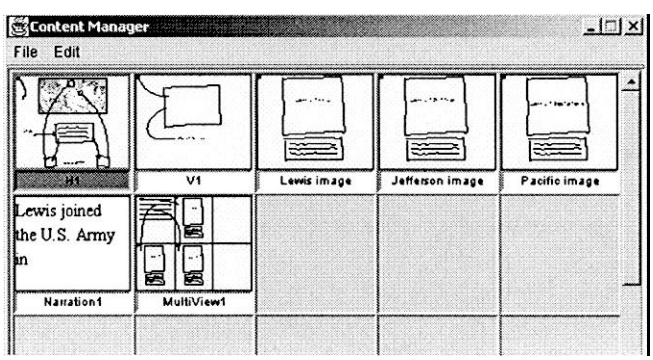

Figure 4. The Content Manager enables a designer to cot, copy, and paste a storyboard, narration, or multi-view and enables a designer to quickly switch among that content for editing.

multi-view editor, a designer sketches an ink stroke (interpreted as a behavioral ink stroke) from a narration's synchronization marker to another storyboard already placed in the multi-view. In Figure 8, because the end of each behavioral ink stroke does not lie within a recognized object in the destination storyboard, the system assigns a default action of navigation to that behavioral ink stroke. However, in Figure 8, a designer has changed the default actions of navigation to medium-speed, dissolve transitions.

During playback, when a narration begins, the system uses a text-to-speech synthesizer to read the text aloud and notifies the runtime engine as each synchronization marker is reached. Once notified, the runtime engine invokes the associated action, if any.

3.1.3 Content Manager: Managing Storyboards, Narrations, and Multi-Views

A designer creates as many storyboards, narrations, and multi-views as desired, and manages that content using the content manager shown in Figure 4. The content manager enables a designer to cot, copy, and paste a storyboard, narration, or multi-view, and enables a designer to quickly switch among that content for editing. The content manager functions and looks similar to the cast window in Director (see [4] for a case study on Director).

DEMAIS does not provide a hierarchical view of the storyboards, as in DENIM and Anecdote. Our focus is on enabling a designer to explore behavior as opposed to an information hierarchy.

3.2 Exploring Behavior: Visual Language Icons, Behavioral Ink Strokes, Annotations, and the Multi-View Editor

To explore behavioral design ideas, a designer sketches behavioral ink strokes between objects in the storyboard or multi-view editor, edits the visual language icons attached to a behavioral ink stroke, or enters textual annotations in the storyboard editor. We discuss the visual language icons first to provide context for the subsequent sections.

3.2.1 Visual Language Icons

The visual language icons enable a designer to edit the triggering source event and destination action attached to a behavioral ink stroke. The attached icons may also serve as a communication aid during a collaborative design meeting.

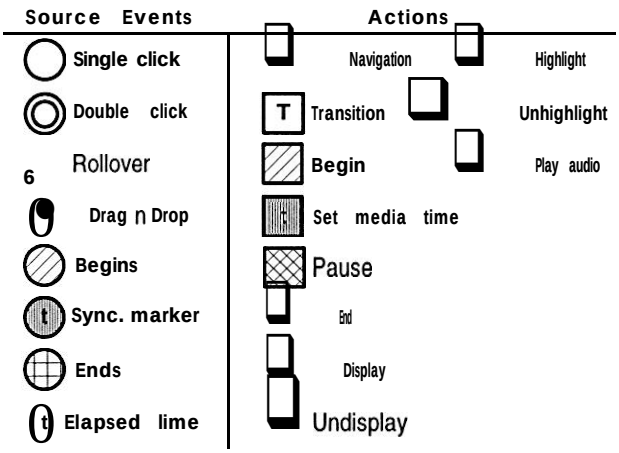

Figure 5. The visual language icons supported in DEMAIS. Above, a shaded icon is color-filled in the application. A 't' in an icon is replaced by an integer value once that icon is dropped on a behavioral ink stroke. For 'Elapsed lime', a designer is prompted to enter the value. while for 'Sync marker' and 'Set media time' the value is set equal to the current media lime of the connected media object.

Using the visual language icons, a designer can quickly compare alternative interaction techniques including drag and drop, rollovers, and single clicks as well as time-based behaviors including elapsed time events and synchronized time points within an audio, video, or narration. Along with those triggering source events, a designer can explore actions including the starting or stopping of an audio, video, or narration, highlighting of an image, display of text, navigation to another storyboard, and more. Figure 5 lists the source events and destination actions currently available in DEMAIS. As we gain more experience with the tool, additional events or actions may be added.

3.2.2 Behavioral Ink Strokes

A behavioral ink stroke is an ink stroke that *connects* two recognized objects or a recognized object and a text object. By *connects* we mean that the ink stroke starts inside one object and ends inside another. For example, in Figure 2, the ink stroke that connects the bottom left rectangle with the rectangle in the center

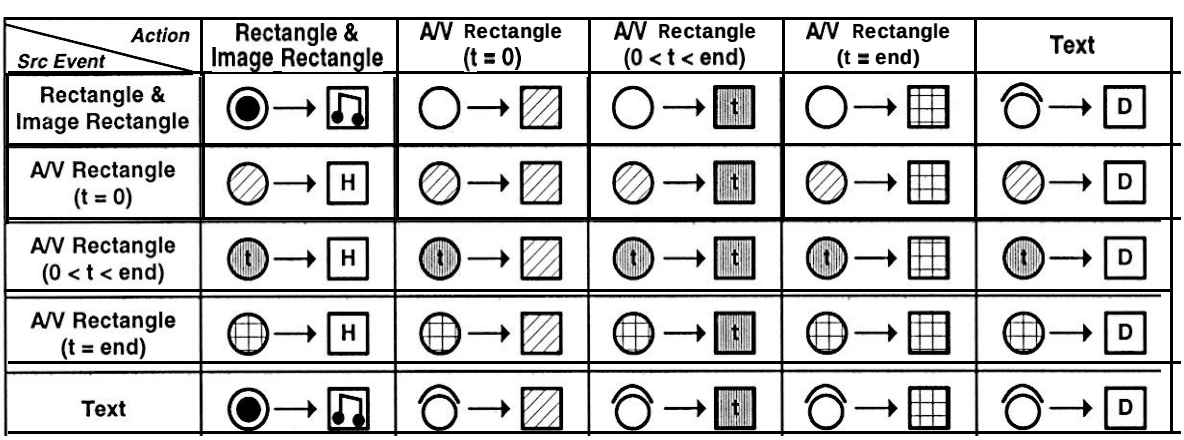

Figure 6. The default source event and destination action table for a behavioral ink stroke. DEMAIS uses the connected object types, and in the case of audio or video (AN), its media time (t), to index into the table and determine the default source event and destination action for a behavioral ink stroke. The source events and actions listed in the table are represented with the appropriate visual language icons.

of the screen is a behavioral ink stroke. A designer defines the majority of behavior in a design using behavioral ink strokes.

A behavioral ink stroke defines an event on the source object (source event) that triggers an action on the destination object (destination action), both of which are depicted using an appropriate visual language icon. The system uses the connected object types (rectangle, image rectangle, audio rectangle, video rectangle, or plain text) and state (audio and video rectangle only) to index a lookup table specifying the default source event and destination action. The default lookup table is shown in Figure 6. For example, when a designer sketches a behavioral ink stroke from a video rectangle with media time (t) to a text caption, the system defaults to a:

- 'Begins' source event, if t = 0
- 'Synchronization marker' source event at t, if 0 < t <end
- 'Ends' source event, if t = end time of the video

and a 'display' action. A designer uses the video's controller on the storyboard canvas to adjust that video's media time. An example of sketching synchronization from a video to a text caption is shown in Figure 7.

The visual language icons representing the source event and destination action are displayed with the behavioral ink stroke. A designer edits the triggering source event as well as the destination action by first ***tapping on an*** icon in the visual language icon panel (shown at the bottom of Figure 2) and then ***dropping*** it near the desired behavioral ink stroke, This tap-and-drop metaphor is more appropriate for a stylus than a drag-and-drop metaphor. Once dropped, the system updates and positions the icon appropriately.

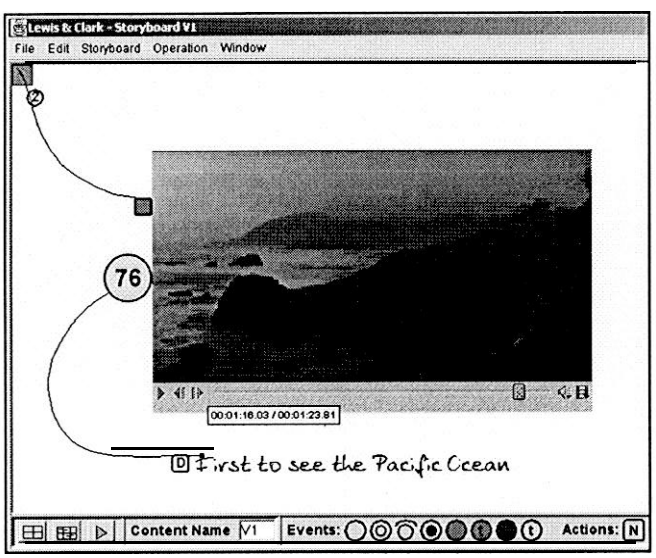

Figure 7. Automatically starting and synchronizing the display of a text caption to a video. To import the video, a designer sketches a rectangular ink stroke, taps the stylus once within it, and selects the appropriate video file. To start the video at playback. the designer sketches an ink stroke from the storyboard start symbol to the video while its at media time 0, in which case the default action is a 'begin.' To synchronize the display of the caption, the designer enters the text, drags the video controller to the desired time point (76), and then sketches an ink stroke from the video to the text. When play mode is entered, the text is erased, the video starts automatically, and then the text is displayed at media time 76. The synchronization marker (76) was enhanced for legibility.

An advantage of this intra-storyboard behavior definition technique is that a designer does not have to create a separate storyboard state for each visual effect as required by SILK. However, in some cases, such as ***spatial collision,*** the need to create separate storyboard states is difficult to avoid. Spatial collision occurs when a designer wants to display two or more objects in the same spatial location at different points in time. To handle this case and others, DEMAIS provides a multi-view editor in which a designer can sketch behavioral ink strokes between objects on distinct storyboards. The multi-view editor is discussed in Section 3.2.4.

3.2.3 *Annotations*

An annotation enables a designer to specify behavior through written text as opposed to sketching a behavioral ink stroke. The motivation for supporting annotations stemmed from our inspection of storyboards gathered during the survey and interview process. We found that a designer often writes a textual statement when referring to a non-visual component such as time or audio. For example, one designer wrote the statement, ***"this*** image *changes* over ***rime"*** on her storyboard to indicate that the application should perform a series of time-based image transitions. To create that behavior using annotations in our tool, a designer would need to create a set of storyboards each containing an image and a textual statement similar to "*After* **5** ***seconds,*** navigate *to storyboard* **X."** A designer could also sketch this behavior using the multi-view editor. The choice is simply a matter of style.

To support annotations, DEMAIS contains a parser based on an LL(1) grammar supporting a wide range of English-like statements such as:

- "*After 5 seconds, start the video*"
- ***"When*** *the* ***audio ends, navigate*** *to* ***storyboard*** *S2*"

When a designer uses an indirect media reference such as ***"the video"*** or "*the* ***audio"*** the system applies heuristics to match that indirect reference to an imported media object.

A designer creates an annotation by entering text anywhere on a storyboard canvas using the storyboard editor as discussed in Section 3.1.1. After the text has been entered, the system parses it. If the text can be parsed without error, then the system marks it as an annotation and passes that annotation to the runtime engine whenever the current design is operationalized. If the text cannot be parsed without error, then it is considered a plain text object and has no interpreted meaning to the system, but it still can be connected to a recognized object using a behavioral ink stroke. A designer can enter as many annotations on a storyboard as needed.

Although a designer must learn the rule syntax for an annotation to work properly, the learning investment is far less than compared to the learning investment for an authoring language such as Lingo [21]. Our annotation language is closer to what a designer already writes on a storyboard than an authoring language is.

3.2.4 *Multi-View Editor*

The multi-view editor enables a designer to attach actions to a narration's synchronization markers or to define behavior between

recognized objects that could not be defined using the storyboard editor alone. The multi-view editor supports up to six storyboards and narrations, and any number of these multi-views can be created. Each multi-view is added to the content manager enabling a designer to switch among them just as he can switch among the storyboards and narrations.

To define behavior in the multi-view editor, a designer sketches an ink stroke from a narration's synchronization marker or an object in a storyboard to another object in a different storyboard or that storyboard itself. An ink stroke made within a single storyboard or narration is rejected by the system. In Figure 8, a designer has sketched three behavioral ink strokes, each from a narration's synchronization marker to another storyboard, and has changed the default actions of navigation to medium-speed, dissolve transitions. When a transition icon is dropped on a behavioral ink stroke, the system prompts the designer to select the speed (slow, medium, or fast) and type (dissolve or wipe) of visual transition to perform. In Figure 8, the system interprets each behavioral ink stroke to mean that when the text-to-speech synthesizer reaches the synchronization marker during playback, a medium-speed, dissolve transition to the specified storyboard should occur, indicated by the attached icon. During playback, the system plays a narration in parallel to the display of storyboards.

3.3 Operationalizing Content and Behavior

A designer operationalizes a design by selecting the play button shown in lower left corner of Figure 2. When a design is operationalized, the system ondisplays the behavioral ink strokes and icons, annotations, and recognized objects and text that have a display action attached to them. If a recognized object is to be undisplayed, then so are all the ink strokes and text objects contained within it, causing them to behave as a group.

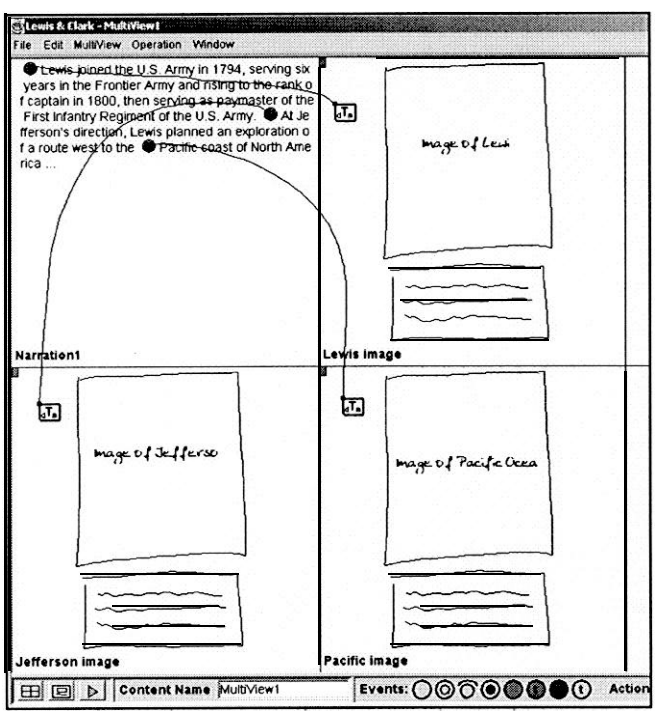

Figure 8. The multi-view editor supports up to six storyboards and narrations. Using this editor, a designer can sketch a behavioral ink stroke from a narration's synchronization marker or an object in a storyboard to an object in a different storyboard or that storyboard itself. Above, a designer has sketched three behavioral ink strokes, each defining a transition action when the corresponding synchronization marker in the narration is reached.

In Figure 2, at playback, the five behavioral ink strokes and icons, the text connected to the start symbol, the rectangular ink stroke in the center of the storyboard and the ink strokes contained within that rectangle are removed. As the designer rolls the cursor over either of the two rectangles near the bottom of the storyboard, the rectangle in the center of the storyboard and its contained ink strokes are displayed. When the designer moves the mouse outside that rectangle, those same objects are undisplayed.

The designer can also experience dragging either of the two lower rectangles to its respective destination on the map. If a rectangle is dropped "close" to its defined area, it snaps to that area and the specified narration begins. Otherwise, the rectangle snaps hack to its original position. When edit mode is re-entered, the undisplayed behavioral ink strokes and icons, objects, and text are re-displayed and reset to their original position and state.

To operationalize a design, the system translates each behavioral ink stroke into an NSync role, and passes those roles along with any textual annotations to the runtime engine. Through these roles and annotations, the runtime engine transforms the current design into a working example.

4. DEMAIS - IMPLEMENTATION

DEMAIS has been in development for approximately 8 months and integrates a number of hardware and software technologies, including a pen-based display, Java and media extensions for it, a layer of media abstraction, and the NSync runtime engine. Figure 9 summarizes how these components work together, excluding the pen-based display. The DEMAIS software can be downloaded from *http://www.cs.umn.edu/~bailey/demais*.

4.1 Pen-based Display

DEMAIS uses a WACOM PL400 pen tablet system [27]. With this system, a designer uses a stylus to sketch directly on the display. The stylus generates input events similar to a mouse device, but causes a designer's interaction with DEMAIS to be considerably different. For example, using a stylus, a designer uses a tap-and-drop metaphor for editing a behavioral ink stroke and performs a single tap (as opposed to a double or right click) on a rectangular ink stroke to bring up a tile browser. For a designer who does not have access to a comparable display, DEMAIS is compatible with a standard CRT and moose device.

4.2 Java, JMF, and Java Speech

DEMAIS was programmed in the Java language [14] and consists of approximately 30,000 lines of code spanning 180 classes. The Java Media Framework (JMF) is used for audio and video support and the Java Speech API coupled with IBM's ViaVoice product is used for text-to-speech synthesis. To generate synchronization markers in the spoken text, the Java Speech Markup Language is used to mark up the text before it passed to the synthesizer.

4.3 Media Element Abstraction Layer

The Media Element Abstraction Layer (MEAL) provides a layer of abstraction between the NSync runtime engine and recognized objects sketched on a storyboard and provides consistent interface wrappers around the JMF and Java Speech components. Each

rectangular object, text, and narration (content object) has a MEAL object associated with it. The separation allows a content object to have one interface suitable for graphical manipulation and another interface suitable for coordinating events and actions with the runtime engine.

When a designer creates a content object; e.g., draws a rectangular ink stroke or creates a voice narration, the system creates an associated MEAL object and registers that object with the Symbol Table (see Figure 9). When a design is operationalized, the compiler retrieves the objects referenced in the rules and annotations from the Symbol Table. As a designer interacts with a sketch during playback, the runtime engine receives events from and invokes actions on a MEAL object, which in turn receives events from the Java toolkit and media extensions and invokes actions on its associated content object. See Figure 9.

4.4 Runtime Engine

The NSync toolkit [5] was re-written in Java and provides the core of DEMAIS's runtime engine. Briefly, the runtime engine is composed of four components:

- *Compiler*. Compiles a rule and passes the equivalent stack-based representation to the Change Monitor and Evaluator.
- Change Monitor. Monitors MEAL object attributes for changes, and once detected, requests that the Scheduler cancel predictions and that the Evaluator compute new predictions for rules dependent on that MEAL object.
- *Evaluator*. Determines the current logic value of a rule expression using a predictive logic. If the expression evaluates to *WBT(x)* or *WBF(x)*, then the predicted time *x*, along with the associated action, are passed to the Scheduler.
- Scheduler. Maintains and schedules the predictions made by the Evaluator. As these predicted times arrive, the associated actions are invoked on the appropriate MEAL objects.

When a designer operationalizes a design, the system translates each behavioral ink stroke into an NSync rule. For example, the rightmost behavioral ink stroke in Figure 2 is translated into the rule "*When the user drags rectangle-A to rectangle-$ speak narration1.*" These rules along with any annotations are compiled and then loaded into the Change Monitor and Evaluator. When the designer returns to edit mode, the runtime engine is cleared and any subsequent play request results in another translation and load process. Figure 9 summarizes this process.

4.5 Limitations

A limitation of DEMAIS is that a designer must enter interpretable text onto a storyboard canvas using the keyboard as opposed the stylus. A designer can always use the stylus to scribble written notes, but these notes cannot be interpreted by the system. However, we are looking at handwriting recognition software as a future enhancement.

Another limitation of DEMAIS is that it does not check for consistency among behavior definitions, possibly causing unwanted results when that behavior is operationalized. For example, a designer could sketch a circular display dependency between two images, causing both to disappear when that behavior is operationalized.

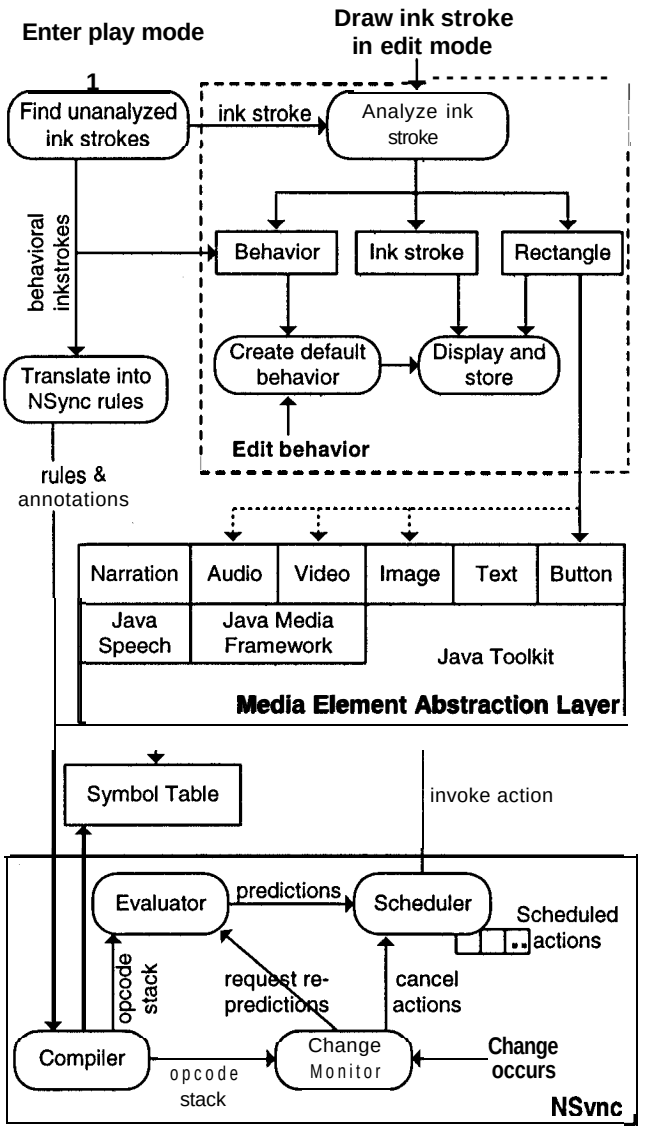

Figure 9. An architectural overview of the edit and play modes in DEMAIS. The dotted lines indicate that a designer can fill a rectangle with an image, audio clip, or video clip after that rectangle has been created.

Finally, although DEMAIS does support layers internally, it does not yet provide a designer with a mechanism for adjusting the layer in which sketched content is drawn. By adjusting layers, a designer can manipulate the overlap among sketched content.

5. INFORMAL EVALUATION AND FUTURE WORK

Two experienced multimedia designers (one who was involved in the design study and one who was not) were given a tour of DEMAIS and allowed to explore its functionality. Both designers expressed positive comments about the tool and were interested in using the tool in upcoming projects. One designer hypothesized that he would like to use DEMAIS during a client meeting in order to produce a working example of an idea as it is being discussed. The other designer was interested in using the tool as a

brainstorming tool and to facilitate the communication of design ideas among team members and clients.

In addition to addressing the limitations discussed in Section 4.5, we plan to:

- *Integrate a sketching interface for 2D animation.* Multimedia applications rely heavily on animation, especially 2D text effects. We are currently investigating the design and integration of a sketch-based 2D text animation tool.
- *Conduct a formal user evaluation.* We want to evaluate the impact of DEMAIS on the design of an interactive multimedia application and compare that result against the impact of other design tools such as pencil and paper and a traditional authoring tool. Our goal is to measure and compare the impact of the design tools along several dimensions such as the degree of early behavior exploration, team communication, design confidence, and quality of the final design,

6. CONCLUSION

Creating innovative behavior is the cornerstone of creating an innovative multimedia application. However, a multimedia designer currently struggles to rapidly explore and effectively communicate behavioral design ideas early in the design process. To address this struggle, we have developed an informal, sketch-based multimedia design tool called DEMAIS. DEMAIS enables a designer to quickly sketch behavioral design ideas using ink strokes and textual annotations, and edit that behavior using an expressive visual language. The benefit of using DEMAIS to sketch behavior early in the design process is that it enables a designer to rapidly transform a mental idea into a *working* example. Using a working example, a designer can directly experience behavioral design ideas, thus enhancing the exploration and communication of those ideas. DEMAIS offers a promising first step towards helping a designer create a more effective, compelling, and entertaining multimedia application.

7. ACKNOWLEDGMENTS

The authors would like to thank the designers who volunteered their time to fill out the survey, meet face-to-face, and share their design stories, processes, and artifacts with us.

8. REFERENCES

[1] Apple Computer. HyperCard: Reference Manual, 1993.

[2] Baecker, R., A. Rosenthal, N. Friedlander, E. Smith, and A. Cohen. A Multimedia System for Authoring Motion Pictures. *Proc. ACM Multimedia,* 1996.

[3] Baggett, P. The Role of Temporal Overlap of Visual and Auditory Material in Forming Dual Media Associations. *Journal of Educational Psychology, 76* (1984), 408-417.

[4] Bailey, B.P. and J.A. Konstan. Authoring Interactive Media. *Encyclopedia of Electrical and Electronics Engineering.* John Wiley & Sons, 2000.

[5] Bailey, B., R. Cooley, J. Konstan, and M. DeJong. NSync - A Toolkit for Building Interactive Multimedia Presentations. *Proc. ACM Multimedia, 1998.*

[6] D.C.A. Bulterman, L. Hardman, J. Jansen, K.S. Mullender, and L. Rutledge. GRiNS: A Graphical Interface for Creating and Playing SMIL Documents. *Seventh International World Wide Web Conference,* 1998.

[7] Caloini, A., D. Taguchi, K. Yanoo, and E. Tanaka. Script-free Scenario Authoring in MediaDesc. *Proc. ACM Multimedia,* 1998.

[8] Faraday, P. and A. Sutcliffe. Providing Advice for Multimedia Designers. *CHI,* 1998.

[9] Faraday, P. and A. Sutcliffe. An Empirical Study of Attending and Comprehending Multimedia Presentations, *Proc. ACM Multimedia,* 1996.

[10] Gould, J.D. and C. Lewis. Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM, 28 (3):* 300-311, 1985.

[11] Gross, M.D. and E.Y. Do. Ambiguous Intentions: A Paper-like Interface for Creative Design. *UIST,* 1996.

[12] Harada, K. and Y. Hara. Anecdote: A Multimedia Storyboarding System with Seamless Authoring Support. *Proc. ACM Multimedia,* 1996.

[13] Hong, J.I. and J.A. Landay. SATIN: A Toolkit for Informal Ink-based Applications. *UIST, 2000.*

[14] Java Technology. http://java.sun.com.

[15] Kant, M., S. Wilson, M. Bekker, H. Johnson, and P. Johnson. Patchwork: A Software Tool for Early Design. *Proceedings CHI'98 Late Breaking Results,* 1998.

[16] Kozma, R.B. Learning with Media. *Review of Educational Research,* 61 (2): 179-211, 1991.

[17] Landay, J.A. and B.A. Myers. Interactive Sketching for the Early Stages of User Interface Design. *CHI,* 1995.

[18] Lewis, M., W. Clark and B. Devoto (Ed). *The Journals of Lewis and Clark.* Houghton Mifflin Co, 1997.

[19] Lin, J., M.W. Newman, J.I. Hong, and J.A. Landay. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. *CHI, 2000.*

[20] Mackay, W.E. and D.S. Pagani. Video Mosaic: Laying Out Time in a Physical Space. *Proc. ACM Multimedia,* 1994.

[21] Macromedia, *Learning Lingo.* Macromedia Press, 1997.

[22] Madsen, K.H. and P.H. Aiken. Experiences Using Cooperative Interactive Storyboard Prototyping. *Communications of the ACM, 36 (4): 57-64,* 1993.

[23] Mayer, R.E. Aids to Text Comprehension. *Educational Psychologist,* 19 (1984), 30-42.

[24] Najjar, L.J. Multimedia Information and Learning. *J. Educ. Multimedia and Hypermedia, 5* (1996), 129-150.

[25] Rieber, L.P. *Getting Interactive with Authorware: Building Simulations and Games.* http://www.nowhereroad.com/authorware/index.html.

[26] Rudd, J., K. Stern, and S. Isensee. The Low vs. High-Fidelity Prototyping Debate. *Interactions, 3* (1): 76-85, 1996.

[27] Wacom Technology. http://www.wacom.com.