

Supporting Innovation: A Behavior-Sketching Tool for Early Multimedia Design

Brian P. Bailey and Joseph A. Konstan

University of Minnesota
Department of Computer Science and Engineering
Minneapolis, MN 55455

{bailey, konstan}@cs.umn.edu

ABSTRACT

To design innovative temporal and interactive behavior for a multimedia application, a multimedia designer needs to rapidly explore numerous behavioral design ideas early in the design process. Because current mechanisms do not adequately support this need, we have developed an electronic sketching tool called DEMAIS. DEMAIS enables a designer to rapidly sketch behavioral design ideas and then run that sketch to experience those ideas. Through its sketching language and layers, DEMAIS reduces the need to create multiple copies of similar storyboards when sketching behavior. In this paper, we discuss the use of layers for sketching, present usability lessons learned from an evaluation of the tool, and discuss how sketching toolkits can better support future applications.

KEYWORDS

Authoring, design, multimedia, sketching, storyboards

INTRODUCTION

A multimedia application enables a user to interactively explore and learn about information through a variety of content and behavior. *Content* refers to the animations, audio, images, narrations, text and video used in the application while *behavior* refers to how the application responds to user interaction and the passage of time.

A multimedia designer faces different design challenges than a user interface or web site designer. Two of those different design challenges are:

- Structuring both static and dynamic content in the temporal dimension and expressing how that structure is affected by user interaction
- Creating user interactions beyond link navigation. To designers, this challenge separates multimedia design from user interface and web site design [4]

However, the key challenge faced by multimedia designers is satisfying a seemingly insatiable demand for *creativity*. This demand comes not only from clients who want something fresh and new, but also from a designer's artistic desire to create something unique [4]. Creativity means more than designing attractive content, it means designing *innovative* behavior that keeps a user actively engaged in an application, which also helps the user better learn and longer retain information [11, 18].

To facilitate innovation, a designer needs to rapidly explore and communicate numerous behavioral design ideas early in the design process [21]. However, in an earlier study of multimedia design practice [4], we learned that a designer *struggles* to explore and communicate behavioral design ideas, limiting opportunities for innovation to occur.

To address a multimedia designer's struggle, we have developed an electronic sketching tool called DEMAIS (DESigning Multimedia Applications with Interactive Storyboards). The benefits of using DEMAIS are that (i) a designer can run the sketch, resulting in a low-fidelity, functional prototype and (ii) it requires low investment, freeing a designer to explore more behavioral design ideas more often. Our work on DEMAIS extends previous work on electronic sketching tools [12, 14] by:

- Developing a visual sketching language enabling a designer to sketch temporal and interactive behavior
- Reducing the need to create multiple copies of similar storyboards when sketching behavior
- Developing a runtime engine enabling a behavioral sketch to be transformed into a functional prototype

Although we have previously discussed DEMAIS from a multimedia authoring perspective [3], this paper discusses how the visual sketching language helps reduce the need to create multiple copies of similar storyboards when sketching behavior, presents usability lessons learned from an evaluation of the tool and discusses how sketching toolkits can better support future applications.

Understanding a Designer's Struggle

A multimedia designer's struggle to explore and communicate behavioral design ideas results from an imbalance between the two most widely used early design

mechanisms: storyboards sketched with pencil and paper and prototypes created with an authoring tool such as Macromedia's Director or Authorware.

Sketching storyboards using pencil and paper requires low investment, but because the storyboards are *static*, it is difficult to assess *behavior*, e.g., deciding whether a voice script should start automatically or in response to a user interaction. On the other hand, building a prototype using an authoring tool is a powerful mechanism for assessing behavior, but it requires a large initial investment. A large investment into an *early* prototype confines the design space to what is doable within that prototype, even though the ideas represented in it are still rough and evolving.

To illustrate the challenge of multimedia design, imagine that you are asked to create an interactive experience regarding the famous U.S. expedition led by Lewis and Clark in the early 1800s [13]. The goal of the application is to create a unique user experience enabling a user to learn about and explore the:

- Biographical information of the expedition members
- Goals and historical context of the expedition
- Path traveled through the U.S. states and territories and the native American tribes encountered along the way
- Tools 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 a complete list, but to convey that this is a typical design example that a multimedia designer often faces.

This rest of this paper is organized as follows. In the next section, we discuss how our design tool differs from other design tools in related domains. In the third section, we describe how a designer uses DEMAIS to create a design. Afterward, we present usability lessons learned from an evaluation of the tool. Then we discuss issues related to the design of the tool, suggest how sketching toolkits can better support applications and provide directions for future work.

RELATED WORK

The user interface community has a rich history in building tools that help designers assess early design ideas. Here we discuss tools in domains related to multimedia design and tools for electronic sketching of interactive designs. In each case, we identify how our tool differs from them.

Tools for Video and Film Design

Video Mosaic [17] enables a designer to experience editing decisions early in the video production process. The tool provides an augmented reality environment containing both a paper and electronic version of a storyboard. A designer can manipulate references in the paper storyboard and then experience the changes in the electronic version.

MAD [1] facilitates the design of a linear presentation by enabling a designer to run the presentation story structure as it exists hitherto. Empirical evidence indicated that the run mode was one of the most beneficial aspects of the tool.

Although these tools have proven useful in their respective domains, they do not support the exploration of temporal or interactive behavior required for an interactive multimedia application, which is the focus of our tool.

Tools for Sketching Interactive Designs

SILK [12] is an electronic sketching tool for early user interface design. Using SILK, a designer sketches a series of storyboards and illustrates transitions among them. When requested, SILK runs the design enabling the designer to experience the storyboard transitions beyond that provided by paper mockups alone.

DENIM [14] is an electronic sketching tool for early web interface design. DENIM provides a zoomable canvas enabling a designer to quickly move among multiple views of the design. Similar to SILK, a designer can sketch navigational links from source objects to destination pages. When requested, DENIM runs the design enabling the designer to experience navigating the information structure. Later work on DENIM [15] has explored sketching global navigation bars and some interactions beyond navigation.

Although user interface and web site design are closely related to multimedia design, the design tools used in 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 explore creative temporal layouts, more expressive interaction, and the use of dynamic media such as audio, video and animation for an application.

Hence, our tool focuses on helping a designer explore temporal and interactive design ideas. We assume that a designer has already produced an acceptable information structure and is ready to explore ideas for content layout and behavior for the application.

Tools for Multimedia Design

Anecdote [9] and MediaDesc [7] are multimedia design tools providing some support for the early exploration of behavioral design ideas. These tools enable a designer to create a set of annotated storyboards and connect them with navigational links or sequence them in time. When desired, a designer can run the design to experience the storyboard transitions. A designer can also explore a design at different levels of abstraction through multiple, synchronized views.

Our approach to helping a designer explore behavioral design ideas is substantially different than these tools. Our approach provides a *sketching* interface as opposed to the more traditional WIMP interface used in these tools. Based on our study of multimedia design practice [4], sketching more closely matches what a designer does in practice.

Tools for Multimedia Authoring

An authoring tool facilitates the implementation of a multimedia application by providing abstractions for visual and temporal layout, media playback and user interaction. Examples of authoring tools include Macromedia's Authorware and Director. Although an authoring tool offers expressive power and flexibility, it is often inappropriate for the early stages of multimedia design. An authoring tool is often difficult to learn and use, is overly focused on building a final form application, and requires too large of an initial investment. These observations were verified in our design study and are consistent with observations made in similar domains [12, 19]. Although our tool helps a designer create functional prototypes, it requires a much lower investment than an authoring tool by more closely matching what a designer does in practice.

DEMAIS

DEMAIS enables a designer to rapidly sketch temporal and interactive design ideas early in the design process. The advantage of using DEMAIS over pencil and paper is that a designer can run the design, resulting in a low-fidelity, functional prototype enabling the designer to experience the behavior represented in that design. The advantage of using DEMAIS over an authoring tool is that it requires a much lower investment, freeing a designer to explore more behavioral ideas more often. A designer only needs to sketch enough of a behavioral idea to effectively communicate it to herself or another person. Using DEMAIS, a designer can:

- Sketch and annotate a set of storyboards
- Edit a storyboard using gestures
- Develop voice scripts and import audio, video and image content into a storyboard
- Sketch temporal and interactive behavior
- Run the sketched behavior

With the ability to rapidly create low-fidelity, functional prototypes, we believe a designer can better explore and communicate behavioral design ideas early in the design process, facilitating the design of more innovative behavior. In the next sections, we describe how a designer uses DEMAIS to create a design and how DEMAIS interprets and runs that design.

Creating a Design

To create a design, DEMAIS offers four tool components: storyboard, voice script and multi-view editors and a content manager. The next sections describe these components and illustrate their use by creating an early design for the Lewis and Clark expedition's interactions with native American tribes. Our design is adapted from one created by an experienced designer using DEMAIS.

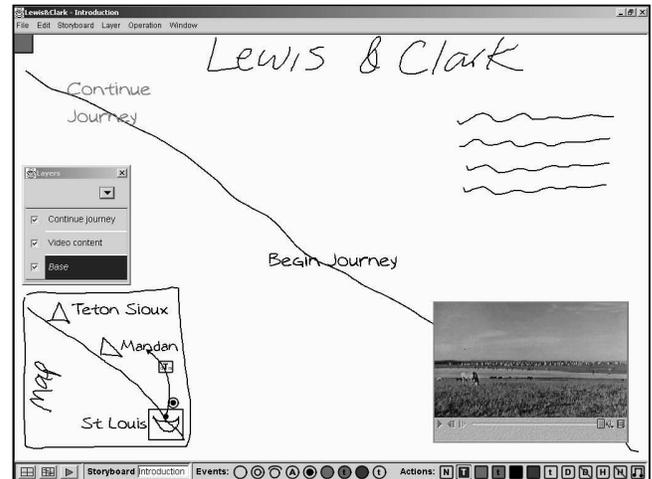


Figure 1. A storyboard showing an introductory screen for Lewis and Clark's interactions with native tribes. The idea is that a user journeys up the Missouri river and interacts with the tribes along the way. The map in the lower left enables the user to navigate to a specific tribe using a drag and drop interface.

Storyboard editor

The storyboard editor enables designers to sketch electronic storyboards similar to how they sketch paper storyboards in practice. The storyboard editor provides a multi-layered canvas on which a designer can sketch ink strokes, enter text and import audio, video and image content. A designer creates an ink stroke by sketching directly on the digitizing display with the stylus. Ink strokes typically make up the majority of a design. A designer enters text by pressing and holding the stylus to the canvas for a short moment and then entering the desired text via the keyboard. Although a designer can scribble text using ink, the system does not recognize that ink as text. A designer imports audio, video and image content by sketching a rectangular stroke, tapping once within it and then selecting the appropriate media file from the file browser that appears. The imported content is scaled and positioned to the bounds of the rectangular stroke. Figure 1 shows an introductory screen for the interactions with native American tribes created with the storyboard editor. The rectangular symbol in the upper left corner of the storyboard represents a "handle" from which a designer can sketch time-based events.

A designer can incrementally add content to a storyboard using layers. Layers in DEMAIS are similar to layers in Adobe Photoshop. However, support for layers was not inspired by graphic design tools; rather, it was inspired by our earlier design study in which we observed designers using transparent paper to demonstrate changes in content due to user interaction or the passage of time.

Using the layer panel shown in the left center of Figure 1, a designer can create and delete layers, switch among layers, re-order layers and toggle the visibility of layers. In Figure 1, the video and continue journey text are in separate layers.

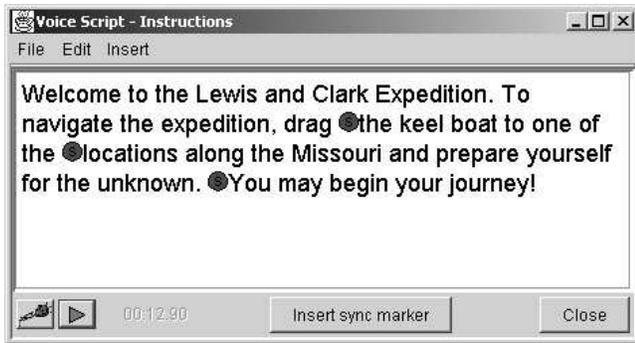


Figure 2. The voice script editor showing a script explaining how to use the interactive map shown in Figure 1. The filled circles represent synchronization markers.

To edit storyboard content, a designer can use gestures similar to those provided in SATIN [10]. However, a designer can only edit content that exists in the currently selected layer. Content in the selected layer is drawn in black while content in the non-selected layers is drawn in a dim gray (for audio, video and images, the borders are drawn in the different colors).

Voice script editor

A multimedia designer considers the development of voice scripts to be an integral part of the early design process [4]. Thus, we developed a voice script editor for DEMAIS. The editor enables a designer to develop one or more voice scripts to accompany a storyboard. To gain a feel for how the script sounds or its approximate spoken length, a designer can have DEMAIS read the script aloud using a text-to-speech synthesizer or record his own voice as he reads the script aloud himself. A designer can also insert any number of synchronization markers into the script. A synchronization marker enables a designer to attach an action, such as a highlight, to a specific point in the script. In Figure 2, a designer is using the voice script editor to develop a script explaining the drag and drop navigation interface shown in Figure 1 to the user.

Multi-view editor and content manager

The multi-view editor enables a designer to sketch behavior between two storyboards or between a voice script and another storyboard. To place a storyboard or voice script into a multi-view, a designer opens the content manager and drags the appropriate thumbnail into the desired cell. A multi-view containing the storyboard and voice script from Figures 1 and 2 along with several behavioral ink strokes between them are shown in Figure 3. The content manager is shown in the lower right of Figure 3.

Interpreting a Design

DEMAIS supports three types of ink strokes: uninterpreted, rectangular and behavioral ink strokes. When a designer sketches an ink stroke on a storyboard or multi-view, DEMAIS attempts to interpret that stroke as a rectangular

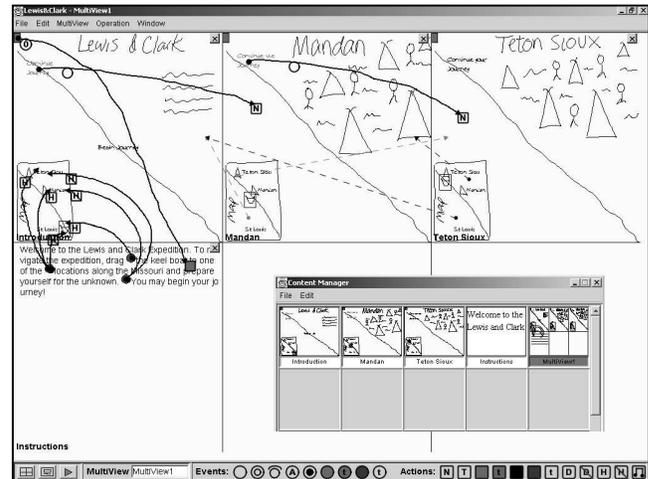


Figure 3. A multi-view with three storyboards and one voice script. The behavioral strokes in the left column cause the script to begin when the storyboard is displayed at playback and the content to (un)highlight as the corresponding markers in the script are reached. The remaining two strokes specify navigation. The content manager is shown in the lower right.

or behavioral stroke, but if unsuccessful, leaves it as ink. The next two sections describe how these interpretations are performed. The third section describes how a designer uses the visual language to sketch behavior without having to create multiple copies of similar storyboards.

Rectangular strokes

When a designer sketches an ink stroke on a storyboard, DEMAIS attempts to interpret that stroke as a rectangular stroke. A rectangular stroke is used to import media content or group a set of existing strokes. To interpret a stroke as rectangular, the system measures the distance from that stroke to a set of pre-defined rectangle classes created with Quill [16]. A stroke is compared using attributes defined by Rubine [20] and available as Java classes in SATIN [10]. If the distance from the stroke to one of the rectangle classes is less than a user-definable threshold, then that stroke is marked as rectangular. If a stroke is incorrectly interpreted as rectangular, a designer can select the stroke and remove the interpretation through a menu or gesture (⌘) command.

Behavioral strokes

When a designer sketches an ink stroke that connects two recognized objects on a storyboard, DEMAIS interprets that stroke as a *behavioral* stroke. A recognized object is text entered via the keyboard or a rectangular stroke which may or may not contain imported media. To locate recognized objects connected by an ink stroke, if any, DEMAIS searches in the current and then lower layers for a recognized object that contains the initial point of the stroke and another recognized object that contains the end point of the stroke. As a result, a designer can sketch behavior between content that exists in different layers.

Source Events	Actions	
 Single click	 Navigation	 Highlight
 Double click	 Transition	 Unhighlight
 Rollover	 Begin	 Play audio
 Drag n Drop	 Set media time	
 Begins	 Pause	
 Sync. marker	 End	
 Ends	 Display	
 Elapsed time	 Undisplay	

Table 1. The visual language icons available in DEMAIS.

When an ink stroke is interpreted as behavioral, DEMAIS assigns a default source event and destination action to it, signified by the appearance of visual language icons near the ends of the stroke. The default source event and destination action are determined through a lookup table indexed by the types, and if appropriate the media value, of the connected objects. A behavioral stroke specifies that when the sketch is run and the specified event occurs on the source object, then the corresponding action will be applied to the destination object or global scope.

In the lower left of Figure 1, the designer has sketched a behavioral stroke from the boat-shaped stroke (boat) grouped by a rectangle to the Mandan tribe name. The source event was set to a drag and drop and the action was set to a transition. When the sketch is run, the behavioral stroke and icons are removed and the designer can interactively drag the boat on the screen. If the boat is dropped on the Mandan (target), the system will navigate to the destination storyboard. Otherwise, the boat snaps back to its original position.

Table 1 summarizes the event and action operators available in DEMAIS. The operators were defined by inspecting several multimedia applications and determining the most common operators used. The icons corresponding to the operators were designed to help maintain a low-fidelity look to the sketch.

To modify an operator on a behavioral stroke, a designer taps on the desired icon from the icon panel (see the bottom of Figure 1) and then taps on the operator to replace. For some operators, the system prompts the designer for additional information, e.g., when a designer assigns a transition action, the system prompts for the destination storyboard as well as the type and speed of the transition.

In a multi-view, an ink stroke sketched by a designer is always interpreted as a behavioral stroke. In Figure 3, a designer has sketched several behavioral strokes from the synchronization markers in the voice script to objects in the storyboard. The strokes specify that when the markers in

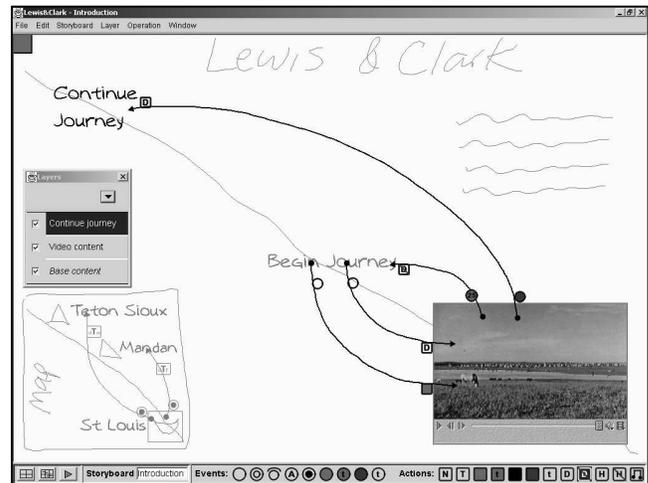


Figure 4. A designer has sketched behavior specifying that when the user selects the *begin journey* text, the video will appear and begin playing. The designer has also sketched behavior specifying that just before the video ends, the *begin journey* text will be removed, and when the video ends, the *continue journey* text will be displayed.

the script are reached during playback, the corresponding objects should be (un)highlighted. Exploring dual media associations is an important part of multimedia design [2].

Sketching behavior without multiple copies of similar storyboards

Using the visual sketching language, a designer can sketch many behaviors without having to create multiple copies of similar storyboards, as required in both SILK and DENIM. For example, in Figure 1, suppose a designer wants the video to be displayed and begin playing back *only after* the user selects the *begin journey* text. Also, suppose that just before the video ends, the designer wants the *begin journey* text to be removed, and when the video ends, he wants the *continue journey* text to be displayed.

To capture this behavior in SILK or DENIM, a designer must sketch three separate storyboards, each representing a successive step in the behavioral sequence. The reason is that any change in content on a storyboard due to an interaction or temporal event can only be represented by creating a *new* storyboard showing that change in content.

Although this approach works, it has at least two limitations. One limitation is that the storyboards are not synchronized, meaning that changes to one storyboard are not reflected in the others. The second limitation is that as more behavior is integrated into the sequence, e.g., showing textual descriptions when the user rolls the mouse over specific objects (known as rollovers), a designer is faced with an explosion of storyboards and links [15].

The visual sketching language in DEMAIS addresses these limitations by enabling a designer to sketch behavior that affects content *within* a storyboard. For example, Figure 4 shows how a designer can capture the same behavioral

sequence in DEMAIS using a single storyboard and our visual language. To cause the video to be displayed and begin playing back only after the user selects the begin journey text, the designer sketches two behavioral strokes from the text to the video. Both strokes are assigned a single click event while one stroke is assigned a display action and the other stroke is assigned a begin media action. To cause the begin journey text to be removed just before the video ends, the designer adjusts the video's controller bar to a time just before the end of the video and sketches a behavioral stroke from the video to the text and assigns an undisplay action. By default, the system assigns a sync marker source event with the media time of the controller bar (25) to the stroke. To cause the continue journey text to be displayed when the video ends, the designer sketches a behavioral stroke from the video to the text and assigns an end media event and a display action.

When the sketch is run, objects with a display dependency (the video and continue journey text) are removed and are not displayed until the specified event occurs. Thus, DEMAIS enables a designer to capture and run this behavioral sequence without creating multiple copies of complete storyboards. Other actions such as (un)highlights can also affect the visual appearance of storyboard content.

Running a Design

A designer runs a design by selecting the play button shown in the lower left of Figure 1. In response, the system removes the behavioral strokes and icons and recognized objects with a display action attached to them. From Figure 4, the five behavioral strokes and related icons, the video, and the continue journey text are removed. The system begins reading the voice script aloud and the boat and tribe names are (un)highlighted as they are referenced in the script (specified in Figure 3). As shown in Figure 5, after the designer selects the begin journey text, the video appears and begins playing. Just before the video ends, the begin journey text will be removed, and when the video ends, the continue journey text will appear in the upper left corner. At any time, the designer can drag and drop the boat to one of the tribe names in the map to navigate to that section of content. When the designer re-enters edit mode, all content and behavioral strokes are re-displayed and reset to their original position and state.

Implementation

DEMAIS integrates a number of hardware and software technologies including a pen-based display, Java and media extensions for it and the Nsync runtime engine. DEMAIS uses a WACOM PL400 [23], which enables a designer to sketch directly on the display using a stylus. For a designer who does not have access to a comparable display, the tool is compatible with a standard CRT and mouse device.

DEMAIS was programmed in the Java language (~30,000 lines of code) and uses the Java Media Framework for

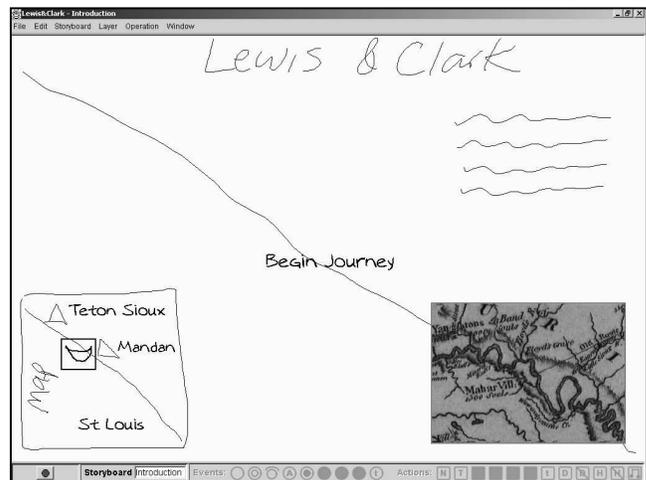


Figure 5. The designer is running the design to experience the sketched behavior. The designer has selected the *begin journey* text causing the video to appear and begin playing. While it plays, the designer is exploring the drag and drop navigation interface. Just before the video ends, the *begin journey* text will be removed, and when the video ends, the *continue journey* text will be displayed, enabling navigation to the next storyboard (first tribe).

audio and video support. The Java Speech API coupled with IBM's ViaVoice product is used for text-to-speech synthesis. To generate synchronization events during synthesis, the Java Speech Markup Language is used to mark up the text before it is passed to the synthesizer. Due to the range of behavior that can be sketched in DEMAIS, neither HTML nor SMIL [22] could be used as the underlying modeling language. Thus, we use our own rule language and runtime engine to model and run a design [6].

FORMATIVE EVALUATION

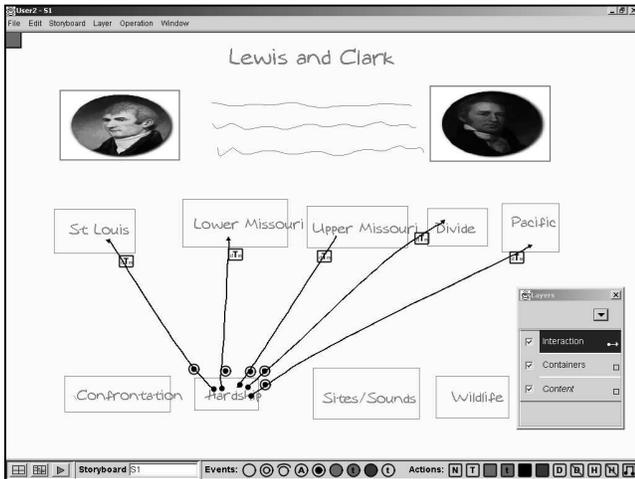
We conducted an evaluation of DEMAIS to assess and improve the usability of the tool, and better prepare it for an upcoming evaluation comparing it against other tools.

Users

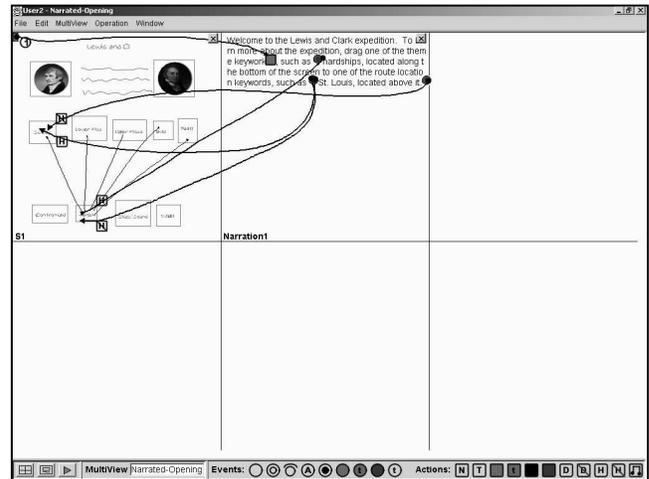
Nine users (two female) participated in the evaluation and were not compensated for their participation. Five of the nine users had professional multimedia design experience. Users ranged from 24 to 40 years in age.

Design Tasks

To assess the different components of DEMAIS, we developed two design tasks. One task asked the user to sketch a drag and drop navigation interface along with one section of content regarding the Lewis and Clark expedition. The navigation interface would allow an end user to drag a theme keyword such as *Hardships* and drop it on a route location such as *Lower Missouri*. In response, the application would jump to the section of content corresponding to the intersection of that theme keyword and route location. To help clarify the use of the navigation interface, the user developed an instructional voice script



(a)



(b)

Figure 6. An example storyboard (a) and multi-view (b) for the drag and drop navigation interface design task. This design task required a user to create three storyboards and two multi-views.

that would be narrated to an end user. The user also sketched one section of content regarding the hardships encountered at the Lower Missouri. We provided some media content, but the user was not required to use it. A subset of completed storyboards and multi-views for this design task is shown in Figure 6.

The other design task asked the user to sketch behavior synchronizing the display of text summaries to the playback of a video and to sketch behavior causing different segments of a video to play back in response to user interactions. Each task lasted about 45 minutes.

Procedure

Upon arriving at the lab, the user was welcomed and given a ten minute overview of DEMAIS. To gain experience with the tool, the user performed a set of practice tasks that included sketching and editing ink strokes, entering text, sketching rectangles and importing media, creating layers and sketching behavior. As the user performed the practice tasks, she was asked to think aloud while the experimenter observed the interaction and jotted down usability issues. The practice session lasted about 30 minutes.

After the practice session, the user was asked to perform one of the two design tasks, chosen in an alternating order. Before starting the assigned task, the user was presented with the experimenter's version of the final design and informed that her final design should look similar. At the onset of the task, the experimenter left the immediate testing area, but was readily available for questions. After completing the task, the user filled out a questionnaire. The entire session lasted about 90 minutes.

Usability Lessons and User Feedback

Based on observing users interact with DEMAIS, answering questions during the tasks, and user discussion

after the evaluation, we learned several usability lessons regarding gestures and stroke recognition. These lessons not only improved the usability of DEMAIS, but should also help improve the usability of other recognition-based interfaces. We discuss these lessons as well as the feedback gained from the questionnaires and user discussions.

Enable a user to immediately recognize the difference between an ink and gesture stroke. We observed that users were often in the wrong mode for the type of stroke they intended to draw. Users often incorrectly pressed (gesture) or released (ink) the modifier button on the stylus before drawing a stroke. Although ink and gesture strokes were drawn using different stroke widths, users did not seem to immediately recognize this difference, and thus, did not immediately recognize a mode error, if made.

Users would benefit from recognizing mode errors at the *start* of a stroke rather than *after* that stroke. This saves the user both time and frustration when a mode error is made. Our approach was to increase the visual disparity between an ink and gesture stroke by drawing the gesture stroke in *red* as opposed to *black*, in addition to using a wider stroke width. Later evaluation indicated that the increased disparity was effective in helping a user immediately recognize a mode error.

Enable a user to cancel a gesture. On several occasions, users realized that they were drawing the wrong gesture shape to achieve the desired action. For example, a user began drawing a cut gesture when a paste action was desired. Once the user realized the error, he lifted the stylus from the canvas and began drawing the new gesture. However, lifting the stylus from the canvas caused DEMAIS to process that stroke, even though the user's intent was to cancel it. Because DEMAIS always invoked the command associated with the closest gesture class, the user would cause an unintended action to be performed.

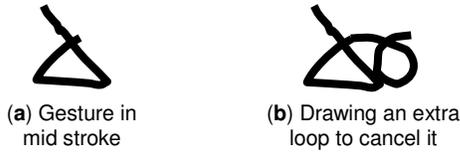


Figure 7. To cancel a gesture in mid stroke (a), a user draws an extra one or two loops at the end (b).

To enable a user to cancel a gesture, we modified our classifier to use a minimum distance threshold such that the distance of the closest gesture class must be less than that threshold. If not, the gesture is considered a no-op. The value for the threshold was determined through empirical analysis. With this modification, a user can cancel a gesture by drawing a loop (or any other disambiguating shape) before lifting the stylus. The loop ensures that the distance from the gesture stroke to the nearest gesture class is *greater* than the minimum distance threshold causing DEMAIS to disregard that gesture. An example of canceling a gesture is shown in Figure 7.

Remind a user of available gestures. Users expressed difficulty recalling the available gesture shapes and their corresponding actions. For example, a user could not recall if a paste action was achieved by drawing the “v” (paste) or “^” (undo) shape. Our approach to addressing this issue is rooted in the fact that most, if not all gestures are shortcuts to menu commands. Thus, to help users recall the gestures available in DEMAIS, we placed an image of each gesture shape beside its corresponding menu command. Figure 8 shows a before and after image of the edit menu. Although DEMAIS uses gestures mainly for editing, we believe this approach would be effective for any set of gestures.

Exploit the spatial location of a gesture. We observed users drawing the copy gesture over the strokes they wanted to copy and the paste gesture at the location they wanted previously copied objects to appear. Several users stated these were the expected behaviors. In response, we modified the action of the paste gesture to position previously copied objects at the location that the gesture was drawn. Although we intend to do so, we have not yet modified the action of the copy gesture. The cut gesture

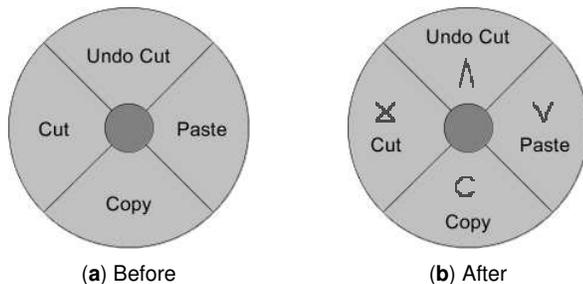


Figure 8. To help users recall the available gestures, their shapes and corresponding actions, a picture of each gesture was placed next to its corresponding command in the edit menu.

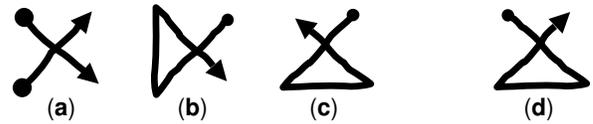


Figure 9. Users drew the cut gesture in multiple strokes (a) and a different orientation (b), initial direction (c), and size (not shown) than the canonical cut gesture (d).

already exploits the spatial location at which the gesture is drawn by erasing the strokes intersecting its bounding box.

Provide feedback for interpreted strokes. Users often forgot whether DEMAIS had classified an ink stroke as rectangular or not. When DEMAIS classified a stroke as rectangular, it would blink that stroke twice. However, the lack of persistent visual feedback allowed the user to forget if this interpretation was made. Because a user can only import media into or sketch behavior involving a stroke classified as rectangular, conveying this information is important. We considered several approaches to this issue including using a different color for the rectangular stroke, embellishing it with an icon, and redrawing it as a “perfect” rectangle. We chose the latter approach as we felt it provided the needed feedback with minimal disruption to the aesthetics of the sketch.

Need better classification algorithms. We observed that users are unwittingly keen at drawing gestures dissimilar to how the gesture classifier was trained. For example, as shown in Figure 9, users drew a cut gesture using more strokes, a different initial direction, and a different orientation than the canonical cut gesture, resulting in incorrect classification. DEMAIS uses a feature-based distance classifier for stroke recognition [20].

To classify variations on a canonical gesture, each one must have its own gesture class. However, this places the burden on the gesture designer as she must anticipate the variations that a user might draw and then train a new gesture class for each of those variations, a process that is cumbersome at best. A better approach is to develop better classification algorithms that are less sensitive to size, speed, rotation and direction variations inherent in end user sketching.

User Feedback

In the questionnaire we asked users to rate how well they understood the concept of sketching with layers, how well they understood the meanings of the visual language icons, and how using DEMAIS would compare to using an authoring tool or pencil and paper for the same design task. The ratings were performed on a five-point Likert scale with (5) being a highly positive rating and (1) being a highly negative rating. The ratings are summarized in Table 2. The last two rows in the table do not sum to nine because some users did not have experience in authoring tools or multimedia design.

Table 2: User Ratings from the Questionnaire

	(5) +	(4)	(3)	(2)	(1) -
Understood layers	3	4	2	0	0
Understood icons	0	4	3	1	1
Comparison to authoring tool	1	2	0	0	0
Comparison to pencil and paper	3	2	1	0	0

Users found the concept of sketching with layers to be reasonably understandable. Seven of the nine users rated sketching with layers to be somewhat or very easy to understand while two users rated it understandable, but not easy. The concept of sketching behavior between content in different layers was the most difficult aspect of layers (and sketching behavior) for users to understand.

Users found the meanings of the visual icons to be moderately easy to understand. Seven users found the icons to be somewhat easy or in-between easy and difficult to understand, while two users found the icons to be somewhat or very difficult to understand. However, because this was the first time the users had seen or used DEMAIS, we did not expect the meanings of the icons to be immediately apparent. Although we do provide tooltips for the icons, a user needs more experience with the tool to gain a better understanding of them.

Users with experience in authoring tools (3 of them) rated DEMAIS as either somewhat or significantly better than using an authoring tool for the same design task. None of these users rated DEMAIS as the about the same or worse than using an authoring tool, which we viewed as an extremely positive result. One user stated that using DEMAIS to explore behavior is “much faster than other authoring tools.” Another user felt that DEMAIS was very easy to learn and use and that with a “simple set of instructions, anyone would be able to quickly create an animated storyboard.”

Similarly, users with experience in using pencil and paper as a multimedia design tool (6 of them) rated DEMAIS as either somewhat or significantly better than using pencil and paper for the same design task. Only one user rated DEMAIS as about the same and no user rated DEMAIS as worse than pencil and paper. One user stated that DEMAIS is “closer to reality and able to show navigation and interaction that paper can’t.” A second user stated that “DEMAIS’s ability to demonstrate an idea makes it easier to communicate the dynamic aspects of an application.” And a third user liked the idea of “having everything on the computer screen and not having to shuffle through a lot of papers.” Overall, users seemed to feel that rapidly sketching temporal and interactive design ideas would be beneficial in the early design process.

DISCUSSION AND FUTURE WORK

In this section we discuss several issues related to the design of DEMAIS, suggest how sketching toolkits can better support future sketch-based interfaces and provide directions for future work.

When a user sketches two or more behavioral strokes from the same source object and assigns the same source event, DEMAIS invokes the corresponding actions in the order that the strokes were drawn, but this order is not conveyed in the sketch. For example, in Figure 4, it is not clear which of the actions attached to the video will be invoked first. Although the order of action invocation is often inconsequential, one user felt that communicating this information was important nonetheless. A few users were also confused whether both or just one of the actions would be invoked in response to the event. We are assessing different approaches to this issue including placing sequence numbers on the strokes, allowing multiple actions to be placed on a single stroke, and having a junction point near the end of a stroke from which multiple action strokes could be sketched.

Although a goal of DEMAIS is to support the sketching of temporal behavior, the tool does not provide a timeline editor similar to Director (see [5] for a case study on Director). Because user interaction is often independent of time, using a timeline editor to express that interaction is often ineffective. DEMAIS allows each dynamic media object imported into a storyboard to maintain its own timeline independent of the storyboard’s timeline (we provide a timeline, but it is not editable). Although this approach allows a designer to sketch many temporal behaviors easily, there are some behaviors that are difficult to sketch. For example, specifying a temporal behavior that causes a text object to be displayed two seconds after a video ends is difficult to sketch. We are exploring extensions to DEMAIS that would enable a designer to sketch more temporal behavior more easily. We are also considering providing a timeline *view* of a design.

The stylus used has a single modifier button (or 1 bit) for selecting modes, however, DEMAIS supports three sketching modes: ink, gesture and behavior mode. When the button is released, the system is in ink mode and when it is pressed, the system is in gesture mode. But there is not a third value to support an explicit behavior mode. Our approach was to mix the ink and behavior modes and define a set of rules for interpreting when an ink stroke is a behavioral stroke. We considered having a designer switch modes through a user interface control, but felt that this would impede the natural flow of sketching. We also considered having a designer sketch behavior in gesture mode, but felt that classifying a gesture as a behavior would be overly difficult and that the persistent nature of behavioral strokes is too asymmetric to the transient nature of gestures. We are exploring the use of speech input to disambiguate the meaning of a stroke, similar to [8].

Based on our experience building DEMAIS, we believe sketching toolkits such as SATIN [10] should support

- Searching for strokes with respect to a layer policy, e.g., search only in the current layer, search all layers or search some subset of layers
- Defining layer styles. A layer style would serve as the default rendering style for graphical objects in that layer. A layer style would override the default canvas style and a specific graphical object could override its containing layer's style
- Mixing ink and widgets. This would ease the challenge of including audio, video and other widgets in a sketch while maintaining correct drawing semantics, i.e., the ink and widgets appear above or below each other depending on the layer and order they were inserted
- Defining how the application responds to an interaction involving a stroke. For example, in response to a user clicking inside a rectangular stroke, the application should be able to define whether that interaction means to select that object, display a file browser or both

In addition to addressing the issues already outlined in this section, we are also planning to

- Enable a designer to sketch 2D animations and integrate those animations into the broader design
- Support multi-hop behavioral ink strokes enabling a designer to sketch a *sequence* of actions for an event
- Enable a designer to define higher-level operators composed from lower level operators.
- Explore the use of a larger display where the artifacts of an entire design can be viewed at once without zooming
- Conduct an evaluation to assess the impact of DEMAIS in comparison to other design tools on the exploration and communication of behavioral design ideas

CONCLUSION

Multimedia design is different than user interface or web design in that it focuses more on temporal and interactive behavior and the use of dynamic media. Based on a study of multimedia design practice, we learned that designers need better tools for exploring and communicating behavioral design ideas early in the design process. To address that need, we have developed DEMAIS. DEMAIS enables a designer to quickly sketch temporal and interactive behavior using a visual sketching language and to experience that behavior by running the sketch. Based on an evaluation of the tool, we have presented new usability lessons for recognition-based interfaces. And based on our experience building the tool, we have suggested how sketching toolkits can better support future applications. By helping designers better explore behavioral design ideas, we hope to facilitate the design of more innovative behavior, leading to more engaging applications for users.

REFERENCES

- [1] Baecker, R., et al. A Multimedia System for Authoring Motion Pictures. In *Proceedings of ACM Multimedia '96*.
- [2] 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.
- [3] Bailey, B.P., J.A. Konstan and J.V. Carlis. DEMAIS: Designing Multimedia Applications with Interactive Storyboards. In *Proceedings of ACM Multimedia '01*.
- [4] Bailey, B.P., J.A. Konstan and J.V. Carlis. Supporting Multimedia Designers: Towards More Effective Design Tools. In *Proceedings of Multimedia Modeling*, 2001.
- [5] Bailey, B.P. and J.A. Konstan. Authoring Interactive Media. *Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons, 2000.
- [6] Bailey, B.P. et al. NSync - A Toolkit for Building Interactive Multimedia Presentations. In *Proc. ACM Multimedia '98*.
- [7] Caloini, A., et al. Script-free Scenario Authoring in MediaDesc. In *Proceedings of ACM Multimedia '98*.
- [8] Cohen, P.R., et al. QuickSet: Multimodal interaction for distributed applications. In *Proc. ACM Multimedia '97*.
- [9] Harada, K. and Y. Hara. Anecdote: A Multimedia Storyboarding System with Seamless Authoring Support. In *Proceedings of ACM Multimedia '96*.
- [10] Hong, J.I. and J.A. Landay. SATIN: A Toolkit for Informal Ink-based Applications. In *Proceedings of UIST '00*.
- [11] Kozma, R.B. Learning with Media. *Review of Educational Research*, 61 (2): 179-211, 1991.
- [12] Landay, J.A. and B.A. Myers. Interactive Sketching for the Early Stages of User Interface Design. In *Proc. CHI '95*.
- [13] Lewis, M., W. Clark and B. Devoto (Ed). *The Journals of Lewis and Clark*. Houghton Mifflin Co, 1997.
- [14] Lin, J., et al. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *Proc. CHI '00*.
- [15] Lin, J., et al. A Visual Sketching Language for Sketching Large and Complex Interactive Designs. In *Proc. CHI '02*.
- [16] Long, C., J.A. Landay and L.A. Rowe. Implications For a Gesture Design Tool. In *Proceedings of CHI '99*.
- [17] Mackay, W.E. and D.S. Pagani. Video Mosaic: Laying Out Time in a Physical Space. In *Proc. ACM Multimedia '94*.
- [18] Najjar, L.J. Multimedia Information and Learning. *J. Educ. Multimedia and Hypermedia*, 5 (1996), 129-150.
- [19] Newman, M.W. and J.A. Landay. Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice. *Designing Interactive Systems*, 2000.
- [20] Rubine, D. Specifying Gestures by Example. In *Proceedings of SIGGRAPH '91*.
- [21] Shneiderman, B. Creating Creativity: User Interfaces for Supporting Innovation. *ACM TOCHI*, 7 (1): 114-138.
- [22] SMIL. <http://www.w3.org/AudioVideo>.
- [23] Wacom Technology. <http://www.wacom.com>.