The Kinedit System: Affective Messages Using Dynamic Texts

Jodi Forlizzi[1,2], **Johnny Lee**[1], **Scott E. Hudson**[1]
[1]Human-Computer Interaction Institute and [2]School of Design
Carnegie Mellon University, Pittsburgh, PA 15213 USA
{forlizzi, johnny.lee, scott.hudson}@cs.cmu.edu

ABSTRACT

Kinetic (dynamic) typography has demonstrated the ability to add significant emotive content and appeal to expressive text, allowing some of the qualities normally found in film and the spoken word to be added to static text. Kinetic typography has been widely and successfully used in film title sequences as well as television and computer-based advertising. However, its communicative abilities have not been widely studied, and its potential has rarely been exploited outside these areas. This is partly due to the difficulty in creating kinetic typography with current tools, often requiring hours of work to animate a single sentence.

In this paper, we present the Kinedit system, a basic authoring tool that takes initial steps toward remedying this situation and hence promoting exploration of the communicative potential of kinetic typography for personal communication. Kinedit is informed by systematic study and characterization of a corpus of examples, and iterative involvement and validation by designers throughout the development process. We describe the tool and its underlying technology, usage experiences, lessons learned, and next steps.

KEYWORDS

Kinetic typography, expressive communications, animation, editing tools.

INTRODUCTION

Communication is a fundamental activity that is central to information exchange and knowledge generation. Since the invention of movable type, typography has become a critical part of communication, and static typography has developed into a well-formalized discipline within communication and graphic design. However, only recently have designers and researchers begun to explore how digital displays might impact the communicative potential of typography.

With the increasing ubiquity of computational power, text is no longer limited to static forms. Typographic forms and compositions can now utilize computational power to change over time. We refer to this as *kinetic typography*. It offers many potential advantages to writers and readers.

To explore these advantages, this paper presents the Kinedit system (Figure 1), a tool that explores the communicative potential of kinetic typography for personal communication. Kinedit has been informed by systematic study of a growing corpus of kinetic typography examples, and iterative feedback by designers during its design and development. We find it particularly useful for personal communication because of its ability to convey emotive content using dynamic written words. We also believe that Kinedit addresses a gap in current tools for building dynamic text compositions – providing a tool for rapid expression with high quality results, but not extremely detailed control (for another end-user kinetic typography tool see also [17]). Finally, we hope that Kinedit will be useful in creating emergent design principles related to kinetic typography, and to generate findings that will lead to future tools supporting tasks such as email and instant messaging.

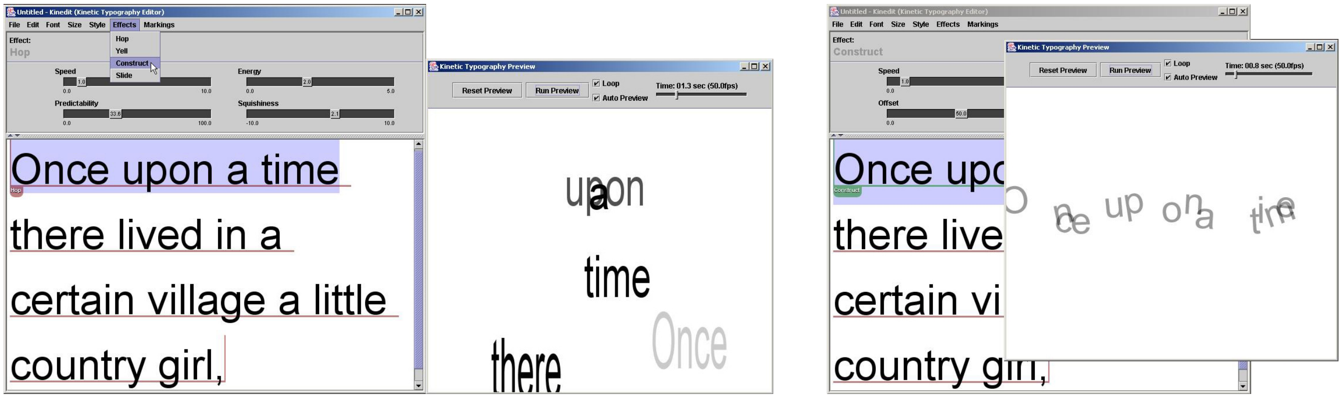
BACKGROUND

The movable metal type invented by Gutenberg was undoubtedly one of the greatest technological innovations of the last millennium. Movable metal presses enabled a new form of expression and communication that could feasibly reach unlimited numbers of readers.

Moving images, or film, were also developed as another kind of expressive medium. In the late 1800s, one of Edison's assistants developed a kinetograph, a camera capable of capturing movement by mechanically advancing film exposures. Over the next century, film would soon become a richly expressive medium that would take advantage of its own unique ability to tell stories, portray emotion, and capture our attention.

Kinetic typography can be understood as a communicative medium that adds some of the expressive properties of film to that of static text. Kinetic typography can be effective in conveying a speaker's tone of voice, qualities of character, and affective (emotional) qualities of texts [7]. It may also allow for a different kind of engagement with the viewer than static text, and in some cases, may explicitly direct or manipulate the attention of the viewer.

CHI 2003, April 5–10, 2003, Ft. Lauderdale, Florida, USA.
Copyright 2003 ACM 1-58113-630-7/03/0004…$5.00.



Effect: Hop

Speed 1.0 (0.0–10.0)

Energy 2.0 (0.0–5.0)

Predictability 33.6 (0.0–100.0)

Squishiness 2.1 (-10.0–10.0)

Time: 01.3 sec (50.0fps)

Once upon a time
there lived in a
certain village a little
country girl,

Effect: Construct

Speed 1.0 (0.0)

Offset 50.0 (0.0)

Time: 00.8 sec (50.0fps)

Figure 1. Kinedit is composed of a main editing window and a preview window. The left image shows the two windows with a hop effect preview in progress. The right image shows the results of applying the "Construct" effect to the same text.

Early origins of kinetic typography developed through two different means, expression and visual perception.

Kinetic type was first employed in the opening credits of films to set the stage by establishing a mood (for examples, see Saul Bass' opening credit sequence for Hitchcock's *North by Northwest* [1] and later *Psycho* [2]). Use of kinetic type is now common in film, television, and computer-based advertising. This is not surprising since dynamic text is particularly effective in expressing mood and attracting the viewer's attention – properties of particular importance in these venues.

A second origin for time-based presentation of text comes independently from psychological studies of perception and reading. For example, [19] studied perceptual effects of a number of text presentations, such as scrolling text. One of the most fruitful of these is a method known as *Rapid Serial Visual Presentation* (RSVP), where text is displayed one word at a time in a fixed focal position [23]. Studies have shown that, because scanning eye movements are unnecessary when using RSVP, it can result in rapid reading without a need for special training. In addition, the RSVP technique provides advantages for designers because it allows words to be treated independently without regard to effects on adjacent text elements. Finally, RSVP enables large bodies of text to be legibly presented on small displays.

Figure 2 illustrates some of the principles of kinetic typography at work. Two different renditions of the same words are shown; each expresses a different emotional tone. As described by Ishizaki [12]:

> *"Suppose it is early Saturday morning. The actor – perhaps a child or puppy – hurries to the door, eager to play outside. One speaker [top] finds sunshine; the other, rain."*

In Figure 2a, the choice of typeface, pace, rhythm, change of scale, and rotation combine to convey a sense of exuberance. In Figure 2b, choice of typeface and a combination of slow and decelerating pace, reduction of typeface weight, and a shrinking motion (analogous to slumping of the shoulders) convey a sense of disappointment.

While the utility of kinetic typography has been recognized for several decades, it still lacks the rich history of either static typography or film. There have been efforts to provide a taxonomy of typographic forms related to interface design [3], as well as examinations of dynamic layouts of static typographic compositions [5, 9]. This work is augmented by a history of work that computationally explores graphical techniques that direct the means of focusing attention on a particular area of a display and increasing the amount of information the screen

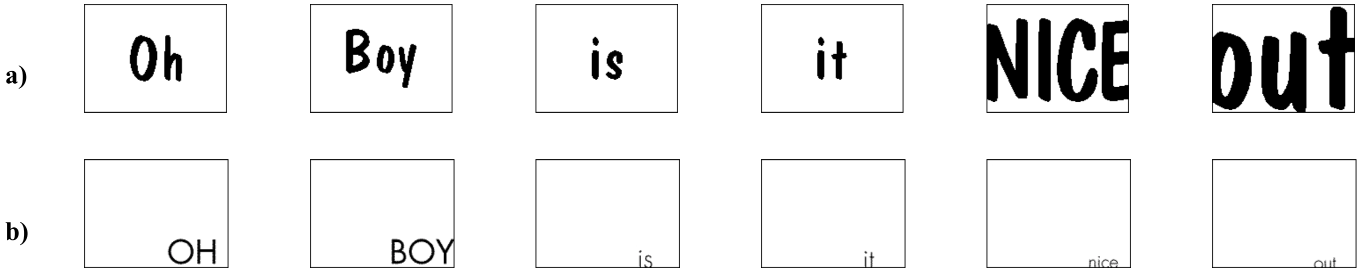


Figure 2. Two examples of the same content expressed with different visual and transitional forms, showing some of the effects of kinetic typography: different emotional tone and distinct characters.

can display [8, 14, 18, 21].

The earliest systematic investigations of time-based presentation of text began in the late 1980s, when designers and technologists manipulated typographic characters by using simulated spring-like physical movements [24]. More recent research has attempted to provide a descriptive language of time-based forms [7, 11, 13, 28]. Our work builds on these efforts to create descriptive schemes for kinetic typography tools. In the next section, we describe the conceptual building blocks for kinetic forms that provide the basis for a library of techniques and effects used in the Kinedit system.

FRAMEWORK FOR THE KINEDIT SYSTEM

The conceptual structures used in the Kinedit system are inspired by those used in classic animation. The craft of animation, advanced by the Disney animators of the 1930s and 1940s, led to a set of design principles and specific techniques which could be employed to turn drawings into compelling and lifelike characters [26]. These techniques include, for example, slow-in slow-out movement, squash and stretch, anticipation, follow-through and secondary action.

Adaptation of these techniques to 3D computer graphics was described by [15] and related to interactive systems in [4]. Finally, based on this mature understanding of the techniques, toolkit level abstractions were created [10, 20]. These abstractions were very easily employed in conventional user interfaces. Notice that the Disney principles do not replace the artist or directly tell one how to create a compelling character. They do, however, provide a vocabulary for talking about animation and a toolbox of proven techniques for using animation.

Overview

Previous work such as [7, 13] has identified three high level communicative goals that have been achieved by kinetic typography. These include:

- Expression of affective (emotional) content,
- Creation of characters, and
- Capture or direction of attention.

In addition, text display on small devices has been identified as a promising area of inquiry [27].

These high level communicative goals can be accomplished using techniques at an intermediate level of abstraction. For example, emotion can often be expressed using techniques for mimicking *tone of voice* and through *analogous motion*. These intermediate level abstractions must be connected to sets of concrete manipulations of properties that can implement them. For example, the prosodic effect of rising pitch at the end of a sentence to indicate a question has been successfully mimicked using an upward movement of a final word, while loudness has been expressed by sudden changes in type size and weight.

In the next section, we illustrate how concrete manipulations in our system have been tied together into reusable abstractions, effects that work to achieve these high-level communicative goals.

Concrete Manipulations

In the Kinedit system, concrete manipulations take a visual form, where they directly impact the appearance of the display, or how visual forms are changed over time. These manipulations are applied to different structural units of text (individual letters, words, phrases, sentences, and higher level units).

Visual forms contribute to the direct visual appearance of the display. These properties can be grouped into four categories: typographic form, type fill, edge, and pose. Typographic form consists of typeface, type style, font size, width of face (extended or condensed), and aspects of leading, or interline spacing. Type fill relates to the actual rendering of letters, and includes color, transparency, and texture and patterning effects. Edge quality concerns the contrast between type and background, focus and blurring of the image, and smooth (anti-aliased) vs. pixelated edges of type. Pose considers position on the screen in x, y coordinates (with possible 3D or 2D effects), rotation, and distortion of the text in space (e.g., shear, curvature of the baseline, or non-uniform scaling).

Transitional forms relate to the manner in which visual forms are changed over time and how the transition between two states of visual form is accomplished. Transitional forms can be varied with respect to their timing (when they occur), duration (how long they take), pacing (uniform speed, or slow-in slow-out, etc.), and path (the set of property values, through which they pass, such as positions).

Effects are created by applying one or more transitional forms between property settings, tying together visual and transitional forms. Figure 3 shows three different transitions (size, transparency, and rotation) applied to the same word, individually and in combination. These combinations of

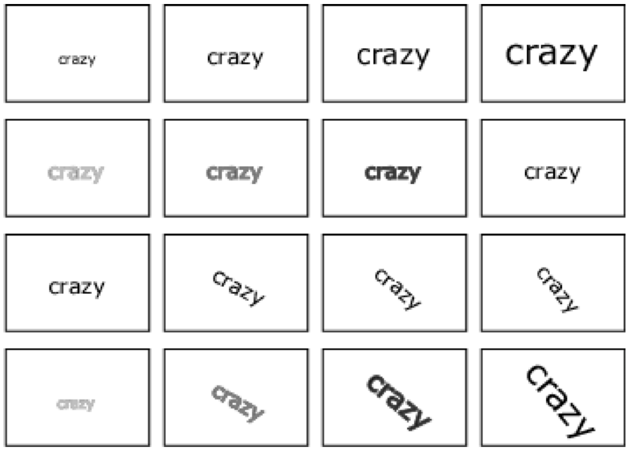


Figure 3. Three different transitional forms applied to the same content, individually and in combination..

visual and transitional forms become the basis for useful and reusable kinetic typography abstractions, or effects that can be implemented in a tool.

Reusable Effects

Sets of reusable effects, or conceptual abstractions, have been developed at an intermediate level of abstraction. These effects have been useful for expressing tone of voice, analogous motion, character effects, and even direction of the reader's attention, matching the high-level communicative goals of kinetic typography discovered in earlier research.

Tone of voice can be understood as variations in pronunciation when segments of speech such as syllables, words, and phrases are articulated. Tone of voice can be roughly divided into two sets of features: paralinguistic features, such as the husky quality of a voice, and prosodic or linguistic features, such as pitch, loudness, and tempo [6].

Design explorations have had difficulty portraying paralinguistic features. However, prosodic features have been effectively conveyed. For example, large upward or downward motions can convey rising or falling pitch. Loudness is used prosodically for a number of purposes. It can affect an entire utterance (establishing its volume, and producing significant effects such as whispers and shouting), a single word or phrase (to accent a word), or a specific syllable (for stress). Loudness can be conveyed by changing the size, weight, and sometimes, contrast or color of text. Tempo characteristics of individual words can be visually dramatized by modifying letter tracking (spatial stretching to indicate a temporally "stretched" word) or through scaling.

Analogous motion uses movements reminiscent of physical actions that convey emotional content. For example, small vibrations that are analogous to trembling can be used to convey affective content with high levels of arousal, such as anticipation, excitement, or anger. When combined with appropriate content, slow rhythmic motions reminiscent of calm breathing can induce feelings of empathy.

While expression of emotion has been a central use of kinetic typography, it is also important to recognize its limitations. In particular, experience has shown that the emotive content expressed by form cannot normally replace or override the emotive content intrinsic in the content. For example, it is not normally possible to use kinetic typography to make a sad story into a happy one. Instead, kinetic typography can reinforce emotive content already present.

In addition to emotional content, kinetic typography has also been successful in portraying characters and dialog. Principles for portraying characters have been adapted from film [25]. These principles include the need to establish identification and re-identification of a character across appearances (and conversely separation of distinct characters). We establish this recognition of a character by use of easily identifiable, distinct and persistent properties for each character. These may include typeface, position, and presentation style.

A second important technique related to character creation is attachment. In film we typically follow (are attached to) one or two characters through time, with other characters appearing only as they encounter the main character(s). Techniques for attachment are still being investigated.

Another property found in many examples of kinetic typography is the ability to capture and direct attention. Here we can draw principles from the scientific knowledge of perceptual and cognitive psychology. For example, perceptual phenomena that have sudden onset tend to induce attentional capture effects [22]. This fact can be used to guide the manipulation of timing and pacing to produce different demands for attention. Similarly, attention can be obtained or directed by using motion. For example, motion along a path can target the viewer's eye to an expected destination. Hence, large sweeping movements are a useful vehicle for explicitly transitioning the user's attention from one location to another. Note that an important potential pitfall of kinetic typography in practical applications can be the demand for too much attention from the reader. As a result, it is important to consider techniques for manipulation of attention in the negative as well as positive, for example, avoiding sudden onsets when attention capture is not desirable.

THE KINEDIT SYSTEM

While frameworks and abstractions are invaluable for conceiving of design ideas, considerable power is also gained by embodying them in a functional tool. Kinedit is an integration of many of these reusable abstractions to allow users to quickly generate rich kinetic typography. Kinedit provides a graphical user interface for entering and editing text that is dynamically animated by an engine designed specifically for animating typographic forms. Both Kinedit and its underlying engine provide manipulations that are representative of the design abstractions discussed in previous sections.

Kinedit Engine

Kinedit is built upon an underlying *engine* for managing and rendering kinetic typography. This engine provides the basic infrastructure for a library of techniques, each of which represents a particular manipulation or class of manipulations. The engine, in turn, works with a graphical user interface for specifying the animation. This section describes the structure of the kinetic typography engine.

The animation engine, described in detail in [16], is an animation API that provides a programmer with a collection of classes for easily representing and executing animation data. The engine serves as the foundation for Kinedit, but can be used as a building block for other kinetic typography applications.

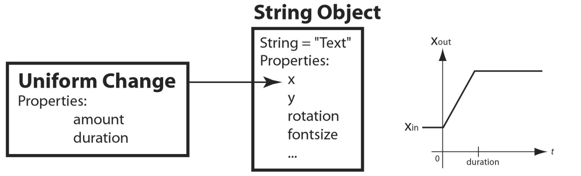


Figure 4. Moving a word horizontally.

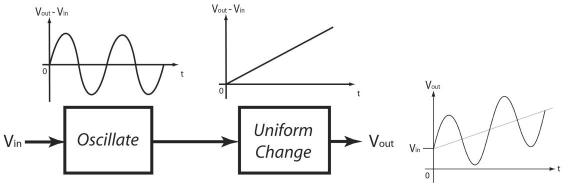


Figure 5. Additive composition.

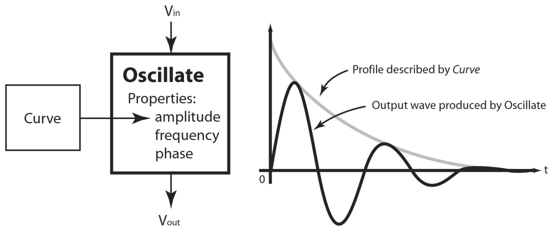


Figure 6. Functional composition.

In the engine, nested hierarchies are used to group text, allowing large and complex animations to be broken up into smaller chunks. Each piece of text or group has the ability to position itself in both space and time. This hierarchy is called a *sequence* tree. This hierarchical structure is similar to the ones used in sophisticated commercial animation packages. Each sequence object in the tree contains a list of *property* objects that maintain the data used to specify time-dependent parameters such as position or size. To cause a property to change over time, a *behavior* is attached to that property updating its value as a function of time and any other necessary parameters. A behavior is often a simple algebraic equation that is naïve of the property it is acting upon (color, scale, etc.) and may be naïve of any other behaviors that may also be acting on that same property. The benefit of this approach is that each behavior can be quite simple, but is also able to produce complex animations through combination with other behaviors.

For example, to move a text object across the screen horizontally, a *UniformChange* behavior is attached to the *x* property of that text object as illustrated in Figure 4. The *UniformChange* behavior calculates a new value of *x* given the value of time and the parameters *duration* and *amount*. Inside this behavior is a simple equation that performs a linear interpolation between the start and end values. In Figure 5, the output of an *Oscillate* behavior is the passed to the input of a *UniformChange* behavior. This is an example of an additive composition where the result is an ascending oscillation. Behaviors can also be attached to the properties of other behaviors. This is called functional composition, as is illustrated in Figure 6. For example, applying a *Curve* behavior to the *amplitude* property of an *Oscillate* behavior produces a pulsation effect. These two methods of interconnecting multiple effects allow a wide range of complex motions to be generated from a small library of simple behaviors.

Because of the extensive control of time necessary in an animation system, the engine provides a similar mechanism for manipulating time itself. A special form of behavior object called a *time filter* is used to modify the value of time as it is passed down the sequence tree when the animation is updated. For example, a *Delay* filter can be applied to subsequence "Scene 1" to start 10 seconds after its parent sequence "Act I" has begun. Time filters may be additively combined to layer the effects of more than one filter, for example *Speed* and *Loop*. Additionally, behaviors can be applied to the properties of a time filter allowing the timing of an animation to also change dynamically.

The mathematical approach that we have developed for generating animation provides a high degree of flexibility and control not available in simple motion path oriented systems. The engine can also leverage the mathematical nature of these animations to automatically generate certain secondary motion effects for the animator. [16] describes a connection between simple cartoon animation techniques (such as squash and stretch) and the derivative of primary motion. These effects can often dramatically improve the appearance and believability of on-screen motion.

Once a designer has crafted an arrangement of behaviors and time filters to generate a desired animation, it can be generalized into a reusable presentation template called a *composite effect.* Similar to a macro, composite effects can be used to quickly animate large amounts of text while providing the user with abstracted high-level controls such as "energy" or "predictability". The author of the effect specifies this list of parameters and determines the flexibility of a particular effect. Currently, programming (guided by examples from designers) is necessary to create these composite effects. However, providing a non-programmatic user interface for generating composite effects is an issue we are currently working on. The collection of these macros, or effects, constitutes a growing effects library that is an important product of this work.

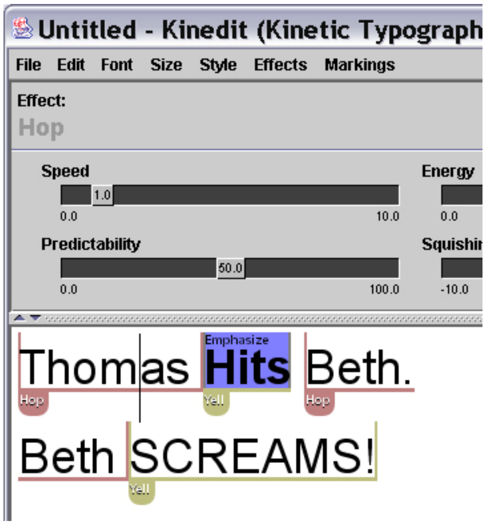
Kinedit Interface

Kinedit, as it has evolved through design and usage, is an editing application built on top of the engine described in the previous section. It allows users to quickly and interactively generate kinetic messages with a graphical user interface. Kinedit is not intended to provide the detailed level of control required by professional visual designers, but provides users lacking extensive experience with kinetic typography the ability to quickly and easily add expressive movement to text.

Kinedit is designed to leverage off of the common text editor application model where a user can enter and edit text into a large window with the keyboard, perform standard editing operations such and cut, copy, and paste on text selections, specify font attributes, and multiple levels of undo. This provides a mature and familiar environment for composing text messages and maintains the ability to transfer large portions of text to and from other applications via copy and paste.

To create an animation, the user highlights the portion of text he or she wishes to animate and selects a particular effect from the Effects menu. This action is equivalent to that of changing font size or font face in a simple text editor. In order to introduce extra information into the text and still stay as close to the text editing metaphor as possible, we make use of an extended interline spacing as illustrated in the lower part of Figure 7 and also employ translucent highlights behind the text. Once an effect has been associated with all or part of the text, Kinedit automatically creates the desired animation tree structure. After a short delay with no editing, the system will (by default) automatically display the resulting animation in a separate preview window. This provides most of the benefits of a WYSIWYG display whereby the full results of changes can be seen immediately, and at a glance. (This can be contrasted to the full WYSIWYG approach used in [17] which can require the user to edit text in complete context, e.g., as it moves.) The preview window also includes basic playback controls such as play, pause, reset, and a time slider, as well as check boxes for looping playback and auto-update preferences. Because of the properties of the underlying engine it is possible to use the time slider to move backwards or forwards across any set of frames at full rendering rate using a "scrubbing" action.

The Effects menu contains a list of pre-designed animation templates that can be applied to any selection of text. These animation templates are merely the composite effects described in the previous section. Though we are continuing to construct more effects, at the time of this writing we provide four distinct effects that were inspired by the corpus of design examples we reviewed. These effects include: *Hop*, where text jumps up from the bottom on the screen and falls back down overlapping in time with new text; *Yell*, which combines shaking and zooming behavior to mimic yelling; *Construct*, where letters converge together to form text in the center of the screen; and *Slide*, horizontally scrolling text that fades in and out. Each effect parses the text into chunks either by line, word, or character depending on the nature of the effect and provides Kinedit with a list of high-level animation parameters such as *speed*, *energy*, and *predictability*. The number, name, and type of parameters vary between effects and are determined by the creator of the effect. The list of parameters provided by the effect is used to generate a simple control UI placed just above the text editing area shown in Figures 1 and 7.


Untitled - Kinedit (Kinetic Typograph
File Edit Font Size Style Effects Markings
Emphasize
Hop
Yell
Hop
Yell

Effect:
Hop

Speed 1.0 (0.0 – 10.0)
Energy (0.0)
Predictability 50.0 (0.0 – 100.0)
Squishi (-10.0)

Thomas Hits Beth.
Beth SCREAMS!

Figure 7. Kinedit Text and Effect settings interface details.

Depending on the variable type of the parameter (Boolean, integer range, floating-point range, string, etc.), the control UI will generate a slider, check box, or text field appropriately. The user can then dynamically change these values using the mouse and keyboard. Often, pushing these parameters to their extremes can result in dramatically different appearances within the same effect. For example, the *Construct* effect (which "assembles" a phrase in the middle of the screen from individual characters) can be adjusted to perform an RSVP style delivery by setting the *construction time* to 0. This causes the word to be assembled instantaneously and the viewer only sees the completed word fade in and then fade out. Therefore, even with a small set of effects and primitive controls, Kinedit is able to achieve a surprisingly wide variety of visual appearances. However, the creator of an effect determines an effect's flexibility.

Each effect may also recognize a set of *markings* that can signal to the effect generator the presence of special words to be processed differently within a specified effect. These markings are available under the *Markings* menu and are applied in the same manner that effects or font attributes are. Again, the effect author determines what markings an effect can recognize and how they will be processed. For example, the *emphasis* marking may signal to an effect generator to emphasize the presentation of a single word in a large phrase. It is the duty of the effect author to decide how this will be executed in the animation when designing the effect. Other markings may mark words to be ignored or to be used for timing purposes either to shift forwards or backwards in time. These markings provide an extra degree of freedom that can be optionally utilized by the user or effect authors.

Because timing is critical to kinetic type, we have also provided a simple shorthand technique for specifying minor shifts forwards and backwards in time without having to leave the keyboard. An isolated comma (,) will insert a 0.25 second delay and an isolated grave accent (`) will go back in time 0.25 seconds. Several instances in succession will concatenate their effects. These let the user quickly make rough timing adjustments easily in the text editor.

Finally, an animated piece can either be saved to a file to be iterated on later or exported to a QuickTime movie file. The pixel dimensions of the movie are inherited from the preview window to ensure what the user sees is what will appear in the video file. Exporting to digital video allows users to share kinetic typography without having to install Kinedit and allows the quality of rendering to be relatively independent of processor speed. Additionally, offline rendering to video offers the ability to create animations that would otherwise be too complex to run in real time.

EVALUATION OF THE KINEDIT SYSTEM

Designers skilled in the creation of kinetic typography have been iteratively involved in the creation and design of Kinedit. The current prototype has been in use for about 4-6 weeks.

Kinedit begins to fill a gap in the tools currently available for generating kinetic messages. It is unlike digital video and animation prototyping tools such as Adobe After Effects and Macromedia Director, because of its focus on text editing, its sensitivity to the natural segmentation of text, and its ability to reuse generalized effects. It is common to see simple effects for presenting time-varying text in video titling tools. However, the presentation techniques in these systems tend to be very rigid and limited in their expressive abilities and rarely accommodate more than a few lines of text.

To better understand how valuable Kinedit might be to designers constructing kinetic messages, we created a 640 x 480 QuickTime movie containing 14 words and employing two effects using the Kinedit system. We gave this movie to a designer with extensive experience in using Adobe After Effects and Macromedia Director for creating kinetic typography and commissioned her to create a loosely comparable reproduction of the piece. After creating the movie using After Effects, she was given a ten-minute tutorial on Kinedit (overview of capabilities, basic operations of menus and windows, and timing and adjustments) and was asked again to create a loosely comparable reproduction of the original piece. While it took the designer approximately six hours to generate the reproduction in After Effects, she was able to achieve similar results in 5 minutes with Kinedit. While this was an anecdotal test with only one designer, and has limitations, the magnitude of the effect – turning an hour of work into less than a minute – is substantial. Kinedit opens up kinetic typography to users without the skill (or extreme patience) previously required. We expect applications like Kinedit, designed specifically for professionals, to affect work practice by dramatically increasing the number of design iterations that can be created in a fixed period of time.

At the time of this writing, we are still performing iterative improvements based on user tests and feedback. The users who have tested Kinedit have generally found it to be an accessible and desirable application. They find the strengths of the tool to include the text editing features and ability to recognize line breaks, the ability to quickly adjust specific effect parameters using sliders instead of entering numbers, and the ability to create many iterations in short time. Designers have also uniformly asked for more control in terms of positioning of specific pieces of text (possibly using a key frame metaphor) and in manipulating timing. Other user driven usability improvements such as reorganization of the main menus are also pending.

DISCUSSION AND FUTURE WORK

Our research has developed a flexible model of kinetic typography, embodied in the Kinedit system and its underlying engine. Kinedit is unlike any of the tools that designers are currently using to create kinetic messages. In particular it allows the creation of many more iterations, and/or supports the use of much longer pieces of text than have been previously practical. We believe that these changes have the potential to significantly change the way we write and use text in digital media.

As next steps, we will be exploring how interfaces including detailed control over pacing, synchronization, positioning and paths for text can be integrated within the current text editing metaphor. In addition, we will be exploring the difficult task of allowing new effects to be created without explicit programming.

ACKNOWLEDGEMENTS

This work was funded in part by the National Science Foundation under grant IIS-01215603. We give special thanks to Jean Zelt for recreating our test piece in After Effects.

REFERENCES

[1] Bass, S. (1959). Opening Credits to *North by Northwest*, Alfred Hitchcock, MGM, 1959. Available on the web at: http://www.twenty4.co.uk/on-line/issue001/project01/clips/nbynwest.rm

[2] Bass, S. (1960). Opening Credits to *Psycho*, Alfred Hitchcock, Universal, 1960. Available on the web at: http://www.twenty4.co.uk/on-line/issue001/project01/clips/psycho.rm

[3] Bork, A. (1983). "A Preliminary Taxonomy of Ways of Displaying Text on Screens." *Information Design Journal,* v3 n3, pp. 206-214.

[4] Chang, B.-W., Ungar, D. (1993). "Animation: From Cartoons to the User Interface", *UIST93 Conference Proceedings,* pp.45-55.

[5] Chang, B-W., Mackinlay, J., Zelleweger, P.T., and Igarashi, T. (1998). "A Negotiation Architecture for Fluid Documents." *UIST98 Conference Proceedings,* pp. 123-133.

[6] Crystal, D. (1975). *The English Tone of Voice: Essays in intonation, prosody, and paralanguage.* New York: St. Martin's Press.

[7] Ford, S., Forlizzi, J., and Ishizaki, S. (1997). "Kinetic Typography: Issues in time-based presentation of text." *CHI97 Conference Extended Abstracts,* pp. 269-270.

[8] Greenberg, S., Gutwin, C., and Cockburn, A. (1996). "Awareness through fisheye views in relaxed WYSIWIS groupware. *Graphics Interface 96 Conference Proceedings,* pp. 28-38.

[9] Hiebert, K. (1992). *Graphic Design Processes: Universal to Unique.* New York: Van Nostrand Reinhold.

[10] Hudson, S.E. and Stasko, J.T. (1993). "Animation Support in a User Interface Toolkit: Flexible, Robust, and Reusable Abstractions." *UIST93 Conference Proceedings,* pp. 57-67.

[11] Ishizaki, S. (1996). "Multiagent Design Systems: Visualization as an emergent behavior of active design agents." *CHI96 Conference Proceedings,* pp. 347-354.

[12] Ishizaki, S. (1997) "Kinetic Typography." Available on the web at: http://www.cmu.edu/cfa/design/kdg/kt/

[13] Ishizaki, S. (1998). "On Kinetic Typography", *Statements, the Newsletter for the American Center for Design*, v12 n1, pp. 7-9.

[14] Kandogan, E. and Schneiderman, B. (1998). "Elastic Windows: A hierarchical multi-window World-Wide Web browser. *UIST 98 Conference Proceedings,* pp. 169-177.

[15] Lasseter, J. (1987). "Principles of Traditional Animation Applied to 3D Computer Animation", *SIGGRAPH 87 Conference Proceedings,* pp. 35-44.

[16] Lee, J.C., Forlizzi, J., and Hudson, S.E. (2002). "The Kinetic Typography Engine: An Extensible System for Animating Expressive Text", *UIST02 Conference Proceedings*, pp. 81-90.

[17] Lewis, J.E. and Weyers, A. (1999). "ActiveText: a method for creating dynamic and interactive texts." *UIST.99 Conference Proceedings*, pp. 131-140.

[18] Masui, T., Minakuchi, M., Borden IV G.R., and Kashiwagi, K. (1995). "Multiple View approach for smooth information retrieval." *UIST95 Conference Proceedings,* pp. 199-206.

[19] Mills, C. and Weldon, L. (1987). "Reading Text From Computer Screens." *ACM Computing Surveys,* v19 n4, pp. 329-358.

[20] Myers, B. A., Miller, R.C., McDaniel, R., Ferrency, A. (1996). "Easily Adding Animations to Interfaces Using Constraints", *UIST96 Conference Proceedings,* pp. 119-128.

[21] Neuwirth, C.M., Chandhok, R., Kaufer, D.S., Erion, P., Morris, J., and Miller, D. (1992). "Flexible diff-ing in a collaborative writing system." *CSCW92 Conference Proceedings,* pp. 147-154.

[22] Pashler, H., (1998). *The psychology of attention*. Boston, MA: MIT Press.

[23] Potter, M. (1984). "Rapid Serial Visual Presentation (RSVP): A Method for Studying Language Processing." *New Methods in Reading Comprehension Research.* Mahwah, NJ: Lawrence Erlbaum, pp. 91-118.

[24] Small, D. (1989). Expressive Typography. Master's Thesis, Media Arts and Sciences, Massachusetts Institute of Technology.

[25] Smith, M. (1995). *Engaging Characters — Fiction, Emotion and The Cinema.* Oxford, England: Clarendon Press.

[26] Thomas, F., Johnston, O. (1981). *The Illusion of Life: Disney Animation.* New York, NY: Hyperion Books.

[27] Wobbrock, J.O., Forlizzi, J., Hudson, S.E., and Myers, B.A. (2002). "WebThumb: Interaction Techniques for Small-Screen Browsers," *UIST02 Conference Proceedings*, pp. 205-208.

[28] Wong, Y.Y. (1995). "Temporal Typography: Characterization of time-varying typographic forms." Master's Thesis, Media Arts and Sciences, Massachusetts Institute of Technology.