SILK: Sketching Interfaces Like Krazy

James A. Landay
HCI Institute, School of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213, USA
Tel: 1-412-268-3608
E-mail: landay@cs.cmu.edu
Web Page: http://www.cs.cmu.edu/landay

ABSTRACT

Current interactive user interface construction tools are often more of a hindrance than a benefit during the early stages of interface design. These tools take too much time to use and force designers to specify more of the design details than they wish at this stage. Most designers prefer to sketch early interface ideas on paper. I have developed an interactive tool called SILK that allows designers to quickly sketch an interface using an electronic pad and stylus. SILK preserves the important properties of pencil and paper: a rough drawing can be produced very *quickly* and the medium is very *flexible*. However, unlike a paper sketch, this electronic sketch is *interactive.* The designer can illustrate *behaviors* by sketching *storyboards*, which specify how the screen should change in response to end-user actions. In addition, it can be semi-automatically *transformed* into a complete, operational interface in a specified look-and-feel.

Keywords

Design, sketching, prototyping, gestures, SILK.

INTRODUCTION

When professional designers first start thinking about a visual interface, they often sketch rough pictures of the screen layouts. Their initial goal is to work on the overall layout and structure of the components, rather than to refine the detailed look-and-feel. Designers use these sketches and other "low-fidelity techniques" [4] to quickly consider design ideas, later shifting to interface construction tools or handing off the design to a programmer. Unfortunately, this transition forces the designer to specify too many details.

Much of the design literature recommends drawing rough sketches of design ideas [1], yet most interface construction tools, such as the NeXT Interface Builder, and even prototyping tools, like HyperCard, require the designer to specify much more of the design than a rough sketch allows. These tools force designers to bridge the gap between how they think about a design and the detailed specification they must create to allow the tool to reflect a specialization of that design.

Another key lesson from the design literature is the value of iterative design. It is important to iterate quickly in the early stages of design because that is when radically different ideas can and should be examined. The need to turn out new designs quickly is hampered by tools that require detailed designs. This over-specification can be tedious and may also lead to a loss of spontaneity. Thus, the designer may be forced to abandon computerized tools until later in the design process or forced to change design techniques in a way that is not conducive to early creative design.

Additionally, research indicates that the use of current interactive tools in the early stages of development places too much focus on design details like color and alignment rather than on the major interface design issues, such as structure and behavior [5]. Wong found that colleagues give more useful feedback when evaluating interfaces with a sketchy look. I surveyed sixteen professional designers from around the world concerning their use of tools in interface design. These designers reported that current user interface construction tools are a hindrance during the early stages of interface design. What designers need are computerized tools that allow them to quickly sketch rough design ideas.

INTERACTIVE SKETCHING

I have developed an interactive tool called SILK [2] that allows designers to quickly sketch an interface using an electronic stylus. SILK then retains the "sketchy" look of the components. The system facilitates rapid prototyping of interface ideas through the use of gestures for sketch creation and storyboards for specifying the transitions between screens. The main advantage of SILK over paper sketches is that it allows the storyboards [3] to come alive and permits the designer or test subjects to exercise the interface in this early, sketchy state. For example, the storyboard in Figure 1 specifies that a dialog box should appear when the user clicks on the button.

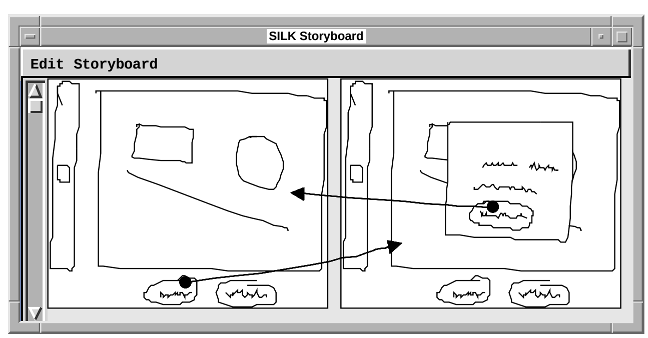

Figure 1: Make a dialog box appear when the button is pressed.

IMPROVEMENTS OVER PAPER SKETCHES

Electronic sketches have most of the same advantages as paper sketches: they allow designers to quickly record design ideas in a tangible form and they do not require the designer to specify details that may not yet be known or important. Electronic sketches also remedy some of the weaknesses of paper sketches.

Editing and Re-use

One of the drawbacks of paper sketches is that they are hard to modify as the design evolves. The designer must often redraw features that have not changed. One way to avoid this repetition is to use an erasable whiteboard. This solution is of no help with the next step when a manual translation to an electronic format is required; this step may need to be repeated several times as the design changes.

SILK allows a designer to easily edit sketched interface designs using simple gestures. SILK's history mechanisms will allow designers to reuse portions of old designs and quickly bring up different versions of the same interface design for testing or comparison. Thus, unlike paper sketches, SILK sketches can evolve without forcing the designer to continually start over with a blank slate.

Design Memory

Another problem with relying too heavily on paper sketches is the lack of support for "design memory." The sketches may be annotated, but a designer cannot easily search these annotations in the future to find out why a particular design decision was made. Practicing designers have found that the annotations of design sketches serve as a diary of the design process, which are often more valuable to the client than the sketches themselves [1]. In addition, paper sketches can be hard to store and organize.

Using SILK, changes made to a design over the course of a project can be reviewed, including viewing the attached written annotations made on SILK's "annotation layer". Electronic sketches also have the advantages of other computer-based tools: they are easy to edit, store, duplicate, and search. Thus SILK will make the "design memory" embedded in the annotations even more valuable.

Interactivity

One of the biggest drawbacks to using paper sketches is the lack of interaction possible between the paper-based design and a user. In order to actually see what the interaction might be like, a designer needs to "play computer" and manipulate several sketches in response to a user's verbalized actions. Designers need tools like SILK that allow them the freedom to quickly sketch rough design ideas and to test the designs by interacting with them.

SILK blends the advantages of both sketching and traditional user interface builders, yet avoids many of the limitations of these approaches. The system tries to recognize user interface widgets and other interface elements *as they are drawn*. Although the recognition takes place as the sketch is made, it is unobtrusive and designers will only be made aware of the recognition results if they choose to exercise the widgets. As soon as a widget has been recognized, it can be exercised.

Next, the designer may specify the higher-level behavior of the sketched elements. For example, what action to perform when a user clicks on a button. This behavior is specified using the sketched storyboards illustrated in Figure 1.

When the designer is satisfied with the interface, SILK will replace the sketches with real widgets and graphical objects; these can take on the look-and-feel of a specified standard graphic user interface, such as Motif, Windows, or Macintosh. The transformation process is mostly automated, but it requires some guidance by the designer to finalize the details of the interface (*e.g.*, textual labels, colors, *etc.*). At this point, programmers can add the application-specific code to complete the application.

STATUS

SILK runs under Common Lisp on both Unix workstations and on the Apple Macintosh with a Wacom tablet attached. It is implemented using the Garnet user interface development environment. The prototype supports the recognition, operation, and transformation of several standard widgets. The current storyboarding mechanism only supports making screen transitions on mouse clicks.

CONCLUSIONS

I envision a future in which most of the user interface code will be generated by user interface designers using tools like SILK rather than by programmers writing the code. I have designed SILK only after examining the problems reported by the intended users of the system. SILK overcomes these problems by allowing designers to quickly sketch an interface using an electronic stylus. Unlike a paper sketch, an electronic sketch will allow the designer or test subjects to interact with the sketch before it becomes a finalized interface. I aim to show that an interactive sketching tool that supports the entire interface design cycle – from early creative design through prototyping, testing, and final implementation – will enable designers to produce better quality interfaces in less time than with current tools.

REFERENCES

1. Boyarski, D. and Buchanan, R. Computers and communication design: Exploring the rhetoric of HCI. *Interactions 1*, 2 (April 1994), 24–35.
2. Landay, J.A. and Myers, B.A. Interactive sketching for the early stages of user interface design. In *Proceedings of CHI '95: Human Factors in Computing Systems,* Denver, CO, May 1995, pp. 43–50.
3. Landay, J.A. and Myers, B.A., "Just draw it! Programming by sketching storyboards," Carnegie Mellon University, School of Computer Science, Technical Report CMU-CS-95-199, November 1995.
4. Rettig, M. Prototyping for tiny fingers. *Communications of the ACM 37*, 4 (April 1994), 21–27.
5. Wong, Y.Y. Rough and ready prototypes: Lessons from graphic design. In *Short Talks Proceedings of CHI '92: Human Factors in Computing Systems,* Monterey, CA, May 1992, pp. 83–84.