

Research Statement

Richard C. DAVIS
School of Information Systems, Singapore Management University
Tel: (65) 6828-1967; Email: rcdavis@smu.edu.sg
Updated on 29 April 2010

Background

I dream of a future where information systems fit organically into human life, where user interfaces feel like a natural extension of the human mind and body. User interfaces will always require learning, but with careful design, they can become invisible as users interact with them. The goal of my research is to demonstrate how to design and build systems that fit naturally into human life.

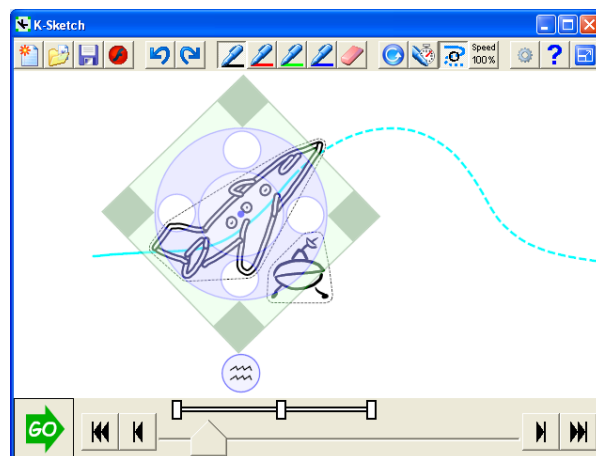
All information systems seek to support some kind of human experience. My general approach is to capture a raw snapshot of that experience from a designer (whether that designer is a trained professional or a novice adapting a system to their own situation). Technology can then help designers evolve these snapshots into working systems. Capturing and manipulating these snapshots often requires the use of rich and responsive input and output modalities, such as pens, multi-touch, and animation, while managing the complexity of these snapshots often requires careful consideration of users' cognitive processes.

An important aspect of my research approach is that I build and deploy working systems. I do this because I think it best to lead by example and because I believe that to truly understand the effects of an interface we must see how it is used to do real work. I have found enthusiastic users for my design tools among educators, creative amateurs and professionals, as well as researchers.

Current research

K-Sketch

K-Sketch is a “kinetic” sketch pad for prototyping animations. More and more people want to create and share animation, but current animation tools are either too restrictive or too complex. K-Sketch allows users to create simple animations quickly and easily with sketching and hand gestures. It also provides a gradual learning curve for creating longer, more complex, or more polished works [2, 4].



K-Sketch: for prototyping engaging animations

Task analysis and user interface modeling were essential to K-Sketch's success. The design process began with field studies that gathered 72 usage scenarios for the tool. Then, conceptual modeling identified 18 possible animation operations, of which 10 were chosen through a process called interface optimization (described later). Only five simple operations are needed to complete about half of the usage scenarios, which helps new users get started quickly and gradually learn more as needed.

K-Sketch has proven its value in laboratory studies and field studies [2, 4]. One laboratory study observed 16 novice animators performing two tasks with K-Sketch and Microsoft PowerPoint. K-Sketch allowed participants to work three times faster and required half the practice time. It also felt significantly faster and easier and imposed much less cognitive load. Similar benefits were seen in another laboratory study that compared K-Sketch to a key frame-based animation tool (The TAB Lite). This study also showed that K-Sketch users needed to do less preparation and less mathematical computation.

In field studies, K-Sketch was used effectively by a user interface designer prototyping animated feedback for a new interaction technique. It was also used by over one hundred high school science students performing learning exercises. I am currently working on a new version of K-Sketch that is more robust and more easily deployable. I am also collaborating with education researchers to deploy K-Sketch in schools where I can carefully observe its effects on student learning and engagement.

StorySketch

A storyboard is a rough mock-up showing a progression of states in a film, video game, or other time-based medium. Designers use them to solicit feedback and to communicate their vision to a production team. Since designers are primarily concerned with their creative task, they often create storyboards with quick sketches that can be rearranged easily. The expressive power of static sketches is limited, however, and designers often create animated storyboards to refine their designs and get more detailed feedback. Animating a storyboard with traditional tools is a laborious process, and the result is not malleable like the original storyboard.

StorySketch will allow designers to quickly and intuitively sketch animated storyboard frames (called flipbooks) and piece them together into longer stories. Unlike traditional animated storyboards, StorySketch storyboards will be easily modifiable, even during design meetings. Individual flipbooks will be created in seconds using a K-Sketch-like interface and assembled into longer stories. Designers will also have the ability to explore alternatives with branching timelines, which can be particularly useful to designers of video games and other stories that do not follow a linear structure.

Game Sketching

While a storyboard can communicate a video game design to a development team, it can be useful for designers to produce a working prototype without the help of developers. For example, amateur designers may not have access to such a team, and even professional designers may wish to collect feedback on a working version of a game before involving developers. Design work has begun on a system similar to K-Sketch and StorySketch that allows designers to make working prototypes of video games by adding user controls and automated behaviors to sketched objects [1].

Interface Optimization

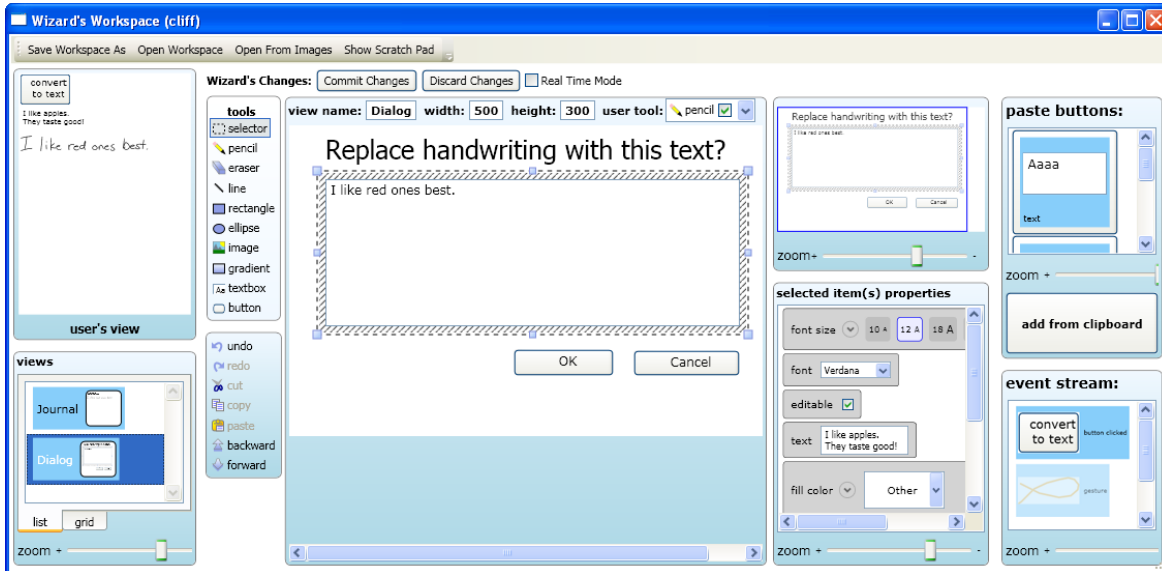
When designing a user interface, a designer must balance many conflicting constraints. The interface must allow users to accomplish tasks quickly, and it should avoid placing high cognitive demands on users, but it is difficult to meet these needs while supporting a wide variety of tasks. Interface optimization is a collection of user interface modeling tools and techniques that help designers balance conflicting requirements for speed, simplicity, and versatility.

Interface optimization was pioneered during the K-Sketch project [2, 4] and expanded during later projects, such as Game Sketching [1]. The process begins with a collection of usage scenarios for an interface. These scenarios are then analyzed to identify good candidates for a user interface model, and each candidate is evaluated against the original set of scenarios. When the analysis is complete, designers have a way to quantify the differences between possible UI models and can even generate visualizations of design tradeoffs. The technique can also help designers weigh the relative importance of concepts within a UI model, allowing them to design a gradual progression of concepts that can facilitate learning.

Wizard of Oz Prototyping

Interfaces that use novel input devices like pens and multi-touch are difficult to design, because few conventions have evolved. When should free-form input be captured? What devices should be used? How will input be interpreted? Will it be recognized automatically? What if recognition fails? Will the system feel natural to users? Every concrete design must attempt to answer these questions, but designers need to check their assumptions against actual user behavior to validate their assumptions. Prototyping is a general technique for handling such situations.

A wizard of oz prototype allows a designer to collect feedback on an interface that appears complete, even though it is being operated by someone “behind a curtain.” My investigation of wizard of oz prototyping began with SketchWizard, a tool that allows designers to simulate real-time responses to sketched input using graphical editing commands [5]. Evaluations of SketchWizard showed that most users do not realize a human being is manipulating these prototypes, even when a designer needs half a minute to prepare responses to users’ input. I am continuing my investigation of the wizard of oz technique with a new system that



The SketchWizard “Wizard’s Workspace” allows a designer to simulate a user interface in real time using graphical editing commands. In the center, a designer is preparing a dialog box that will be shown to a user during a test. The user’s current view is in the upper left.

will capture input on a variety of devices, interpret it with recognition engines, and generate automated responses with simple designer-created macros. This will allow designers to simulate the final interface more accurately and capture data that can be useful to a recognition engine development team.

Future research

In the short term, I plan to continue the investigations I have outlined in previous sections, which deal primarily with the early stages of design. In the long term, I wish to focus more on the later stages of system deployment. Following are some specific research directions that I would like to pursue.

Enabling the Next Generation of Applications: Many computer systems now support input with novel devices, such as pens and multi-touch surfaces. Also, many researchers have developed novel input [6] and analysis techniques that take advantage of this input. So why are there so few applications that take advantage this input? I believe that breakthrough applications will only be found through collaborative experimentation by technologists, designers, and users. Designers and technologists work together to help users solve real-world problems. Through iterations of this process, each group informs the other and good solutions evolve. I have proposed an extensible graphical editor that will allow more rapid iteration of this kind, allowing us to more quickly discover the next generation of applications [3].

Managing Ideas Throughout Their Life Cycle: An artificial wall currently exists between users’ rough sketches and the finished artifacts produced from them. This boundary is difficult to cross, and it causes ideas to become fragmented and unmanageable. For example, animators often make character sketches in a

notebook, but if they wish to show their clients an animated storyboard, they will scan their drawings and animate them in a complex tool like Adobe AfterEffects. The character may be modified in AfterEffects, but it is hard to bring this sketch back into the notebook where it can be browsed alongside other character sketches. K-Sketch blurs the boundary between rough sketches and the animations produced from them, making it possible to move seamlessly from brainstorming character ideas to working out character motions and back again. I plan to extend this idea and build an electronic notebook that helps users managing rough notes or sketches and move them gradually to and from more finished forms such as animations, presentations, and web sites.

Selected publications and outputs

- 1 Colwell B, Davis RC, Landay JA, A study of early stage game design and prototyping. Technical report UW-CSE-08-10-03, Computer Science and Engineering Department, University of Washington, Seattle, WA, 2008. <ftp://ftp.cs.washington.edu/tr/2008/10/UW-CSE-08-10-03.PDF>
- 2 Davis RC, *K-Sketch: A Kinetic Sketch Pad for Novice Animators*, Unpublished doctoral dissertation, EECS Computer Science Division, University of California, Berkeley, CA, December 2008. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-171.html>
- 3 Davis RC. Playing with Recognizers: A Call for an Extensible Editor *Proceedings of the CHI 2010 Workshop on Designing Sketch Recognition Interfaces* Atlanta, GA, April 2010.
- 4 Davis RC, Colwell B, Landay JA. K-Sketch: a “kinetic” sketch pad for novice animators. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp 413-22. Florence, Italy, April 2008.
- 5 Davis RC, Saponas TS, Shilman M, Landay JA. SketchWizard: Wizard of Oz prototyping of pen-based user interfaces. *Proceedings of the ACM Symposium on User Interface Software and Technology*, pp 119-28. Newport, RI 2007.
- 6 Odell DL, Davis RC, Smith A, Wright PK. Toolglasses, Marking Menus, and Hotkeys: A Comparison of One and Two-Handed Command Selection Techniques. *Proceedings of Graphics Interface*, pp 17-24. London, Ontario, Canada, May 2004.