A tool-based interactive drawing environment

Robert St. Amant and Thomas E. Horton
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534
stamant@csc.ncsu.edu, tehorton@eos.ncsu.edu

ABSTRACT

Graphical user interfaces rely heavily on the tool metaphor. In most drawing systems, for example, functions are organized as they might be on a workbench; buttons associated with drawing modes for lines or rectangles are called line-drawing or rectangle-drawing tools; etc. Despite the similarities, however, there remain many differences between software tools and physical tools. This paper gives a concise account of tool use in general, and describes a drawing application, called HabilisDraw, that relies on a detailed correspondence to physical tool behavior.

KEYWORDS

Metaphors, tool use, interface design

INTRODUCTION

The metaphor of tool use for describing the interaction between a human and a computer is pervasive in user interface design. It is surprisingly difficult, however, to define the concept of tool use precisely. Examples of tools in software vary widely. At one extreme, a tool can be as simple as an icon in a drawing application that sets a specific mode (e.g., "Hold the mouse down on the ellipse tool…") At the other extreme we find hundreds of systems in the HCI literature characterized as "tools" that have little more in common than being interactive software. The goal of developing a detailed definition of tool use is not simply for linguistic reasons, or to separate interactive software arbitrarily into tools and non-tools; rather, we believe that a better understanding of the characteristics of tools and tool-using behavior will lead to the design of more effective software.

Researchers in a variety of fields, including social psychology, anthropology, and HCI [2, 3, 4, 5], have examined the concept of tool use. These accounts identify specific properties that make tools effective: tool use involves direct action with immediate feedback, it is goal-directed behavior, and it is associated with intelligent use of space, among many other characteristics. Other important properties, however, have been less often considered in interactive software.

Tools are persistent artifacts. In physical environments, tools are persistent, structured objects with specialized behaviors and properties that can be manipulated, not global modes or transient dialogs as is common in software. One can adjust a wrench or a T-square, for example, to a particular job, or set up a jig for repetitious actions. Persistence leads to other properties as well:

- *Tools encapsulate information as well as behavior.* In the physical world, information is localized to specific tools, in their settings, adjustments, or markings; information can be compared across tools and can persist across the use of different tools.
- *Tools can combine or interact with one another for effect.* The interactions of physical tools are richer and more dynamic than in software. For example, a draftsperson can draw a circle without a specialized compass tool by combining the effects of a pushpin, string, and pencil. Each of these tools has independent uses, rather than only a subordinate relationship to another tool.

The applicability and use of a tool are determined by its ecological properties. For example, a car mechanic who lacks a hammer can use any sufficiently solid, heavy object that comes to hand. The ecological nature of tool use leads to other properties:

- *Tool use can be opportunistic.* Tools can be used for purposes not intended by their designers, as above.
- *Tools provide rich cues about their appropriate use.* The affordances of a tool become obvious in its use; the hammer is almost a canonical example. Comparable (though less compelling) examples of affordances can be found in current software environments [1, 6], but cues for usage in interactive software are much weaker than they might be.
- *Tool use involves establishing and exploiting constraints between the user and the tool, the user and the environment, and the tool and the environment.* Establishing appropriate constraints will enhance tool use, e.g., bracing oneself for a stronger hammer blow, or clamping materials for appropriate resistance.

We have developed a prototype drawing application, called HabilisDraw, to explore the potential of a tool-based software environment that explicitly incorporates all of these

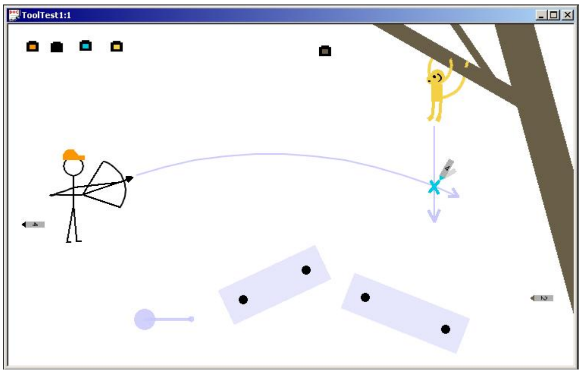

Figure 1: The HabilisDraw environment

properties. Not all of the tools in HabilisDraw are unique; the novelty of this work lies instead in its conceptual focus on tools as first-class artifacts, distinct from the general environment and from created objects.

HabilisDraw tools share some commonalities in their use: they may be moved or adjusted with the right mouse button without affecting other tools or drawn objects; the left mouse button activates a tool's function. The properties of a selected tool or drawn object can be edited in a properties window, and dynamic information associated with a tool (e.g. a ruler's length, a compass's position) becomes visible on the tool when it is in use. The following tools are currently supported, with more in development:

Pens and Ink Wells. A pen can be dragged around the page without interacting with other tools or objects. Once activated the pen will track the cursor. A pen can be dipped in ink wells of different colors, and multiple pens can be left on the page. Only one pen can be active at one time, however. Other classes of tools (rulers and compasses, currently) impose constraints upon the pen. When the pen's tip is properly positioned, the helper tool's active edge or surface is highlighted, and the pen's movement will be constrained by the helper tool.

Pushpins. Pushpins are used as handles for adjusting the position of objects. For lines, they modify length and angle, as though the line were a piece of elastic stretched between two pins stuck into the page. Pushpins can be used to fix an object's spatial properties; they may also be free-standing, to act as point guides for the placement of other tools.

Compasses. A compass tool constrains a pen's motion such that it follows a circular path around the center of the compass's base. Compasses can be used to generate circles or arcs, filled or unfilled. The compass tool consists of a circular base and a circular activation site connected by a narrow arm, interacting as expected with pen and pushpin.

Rulers. A ruler is simply a stretchable and rotatable rectangle that can be used as a straight edge, though in a future release they will include markings so that rulers may more easily be used for measurement. Rulers interact with compasses, pushpins, and other rulers, as well as with pens.

Lenses. The lens is a basic magnifying glass analog, a version of a MagicLens, which shows a detailed view of part of the page. The magnifying and positional properties of the lens can be varied as appropriate.

Figure 1 diagrams a familiar physics demonstration, created using HabilisDraw. The tree and the monkey's body and trajectory were drawn using a pen tool by itself. The circles forming the monkey's head and eye, and the arcs forming his arms, ear, and tail were all created using a pen tool in combination with a compass, as was the arc showing the trajectory of the arrow. The arrow itself, and the hunter's cap, head, and body were created using a pen and a pen-compass combination. The angles forming the hunter's arms and legs were drawn using a pen tool with two rulers set-up as jigs. Finally, the bow was drawn using a pen tool, two rulers and a compass, all in combination.

We have carried out a partial task analysis of HabilisDraw as well as an initial formative evaluation with six users. After exploring the application for a few minutes, users (especially those with more computing experience) did become proficient in its use. Comments on most of the individual tools were favorable. For example, the compass tool was popular; users found it easy to predict the size of the circle or arc that would be created. Most users also liked the option of having multiple instances of a tool, each with different settings. Some of the difficulties that users encountered dealt with the use of two mouse buttons, undersized selection targets, and lack of feedback for some actions. Our main evaluation findings suggest that some refinement is needed in the details of the tools' visual and behavioral properties, and that more tools would increase the power of the system. Results to date have been promising and development is continuing.

ACKNOWLEDGMENTS

This effort was supported by the National Science Foundation under award 0083281.

REFERENCES

[1] William W. Gaver. Technology affordances. In *Proceedings of the CHI'91 Conference*, pages 79–84. Association for Computing Machinery, 1991.

[2] Saul Greenberg. *The computer user as toolsmith : the use, reuse, and organization of computer-based tools.* Cambridge University Press, Cambridge, UK, 1993.

[3] Edwin Hutchins. *Cognition in the Wild.* MIT Press, Cambridge, MA, 1995.

[4] Charles M. Keller and Janet Dixon Keller. *Cognition and tool use: The blacksmith at work.* Cambridge University Press, Cambridge, UK, 1996.

[5] Gün R. Semin. Cognition, language, and communication. In Susan R. Fussell and Roger J. Kreuz, editors, *Social and Cognitive Approaches to Interpersonal Communication*, pages 229–258. Lawrence Erlbaum, Mahwah, NJ, 1998.

[6] Robert St. Amant. User interface affordances in a planning representation. *Human Computer Interaction*, 14(3):317–354, 1999.