Power Tools and Composite Tools: Integrating Automation with Direct Manipulation

John M. Daughtry and Robert St. Amant
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
+1 919 515 7938
{jmdaught, stamant} @eos.ncsu.edu

ABSTRACT
This paper describes a drawing system that incorporates two novel interaction techniques based on analogies to physical tools. Power tools add limited autonomy in the form of rotators and movers for automated circular, linear, and Bezier-curve movement. Composite tools are user-constructed combinations of existing tools to create new functionality.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms
Human Factors.

Keywords
Drawing environments, tools.

1. INTRODUCTION
In user interface design, a tension exists between direct manipulation and automation, especially in the form of agents [2]. Direct manipulation emphasizes physical actions for selection of objects and operations, user control of the environment (which generally remains unchanged otherwise), and immediate feedback on user actions [6]. Agents, on the other hand, attempt to automate some actions on behalf of the user and may dynamically adapt the system's responses to the user's choices [3]. There has been some debate in the HCI community about the tradeoff between giving the user more functionality, via agent automation, and taking away direct control over low-level details of the behavior of a system.

We have recently begun to explore the concept of *tools* as a unifying concept for integrating automation into a direct manipulation environment. In general usage, "software system" and "tool" are interchangeable terms. One might describe a program that counts the words in a document as a tool (and this functionality is indeed accessible from the "Tools" menu in the most popular word processing software.) At the opposite end of the spectrum of sophistication, we might find large CAD systems described as tools for engineers. Our usage of the term tool will be much more specific, as suggested by the summary in Table 1; we are attempting to develop a detailed account of interface tools as metaphorical extensions of physical tools, thus exploiting our natural abilities as tool-using beings [4].

*IUI'03***,** January 12–15, 2003, Miami, Florida, USA.
ACM 1-58113-586-6/03/0001

In this paper we extend our earlier work on simple interactive tools for a drawing environment [4][5] to incorporate the concepts of power tools and composite tools in the interface.

Power tools can significantly simplify or reduce the effort for a task in the physical world. A professional carpenter may use a screwdriver attachment for an electric drill in order to drive screws more efficiently, switching to a manual screwdriver only when special care must be taken in the task of fastening pieces together. A hobbyist potter might "throw" pots without the help of specialized tools, but professional potters generally use a wheel, either muscle-driven or electrically powered, thereby setting up an autonomous process that continuously spins the pot on behalf of the potter. We can extrapolate some of the features of such physical power tools to software power tools, as follows.

- *Visible actions.* The effect of a power tool is achieved through a process visible to the user, mimicking the behavior

Table 1. Characteristics of basic tools and tool use

Tools are persistent artifacts: tools are replicable and reusable; tools encapsulate information as well as behavior; tools can combine with one another for effect.
Tool use exhibits significant procedural structure: direct rather than indirect action; amplification; goal-directed activity; effective behavior.
Tools fall into a natural taxonomy: effective tools produce a persistent effect on materials; instruments gather information; constraining tools constrain materials or the environment; demarcating tools spatially structure the environment.
Tool applicability is determined by ecological properties: tool use can be opportunistic; tools provide rich cues about their appropriate use; effective use of space is important; tool use exploits constraints between the user, the tool, and the environment.

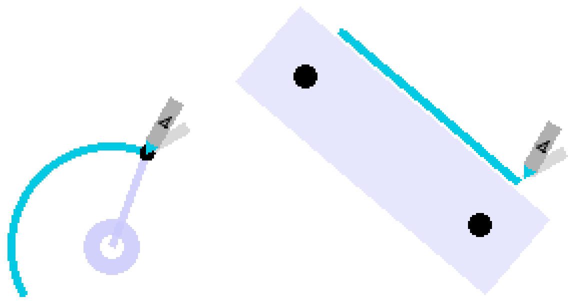

Figure 1: A compass tool from HabilisDraw 1.0.

of a conventional direct manipulation tool.

- *Direct (but optional) user guidance of tool behavior.* A power tool can be controlled either entirely by an autonomous process or directly by the user.
- *Restricted and encapsulated functionality.* Power tools are not generic; rather, they are activated and applied for very specific purposes.
- *Predictable autonomy.* Restrictions on power tool functionality extend to when their actions are allowed.

Composite tools are based on a related analogy to physical tool use. A professional mechanic may maintain a wide range of socket wrenches for installing and removing different types of bolts. The stored wrenches are not complete in themselves, however; they are interchangeable parts that can be composed to meet the needs of specific situations. Some of these tool compositions are so useful that they are retained as complete tools (e.g., many mechanics have multiple sets of socket-wrenches and leave some of the parts attached at all times, such as one handle with an extender and another without.) Other examples of composite physical tools are screwdrivers with interchangeable heads and drills with multiple bits and attachments. The parts of these tools are the result of breaking down a single complex tool into components that can be used individually or combined for new effects.

In our original account of software tool use, we observed that tools can be combined with each other to meet the needs of new problems [4]. We extend that idea here by regarding novel tool combinations as tools in themselves, with the same properties of persistence, procedural structure, opportunistic application, and so forth. We call these combinations composite tools. Again we can identify a number of properties of physical composite tools that apply to their software analogs:

- *Tool combination.* Composite tools are produced by combining existing tools into new tools with new functionality.
- *Restricted and encapsulated functionality.* Composite tools are not generic; rather, they are created for very specific purposes.
- *Predictable autonomy.* Restrictions on tool functionality for each component of a composite tool extend to the composite tool. If one component is a power tool the composite tool is as well.

In our experience, implementing power tools and composite tools in a domain requires a detailed physical or spatial metaphor to support these properties. The system described in this paper is a drawing environment, which has a natural relationship to a physical process. Other domains that we are beginning to explore involve more abstract computations; these need work to impose an extended tool metaphor.

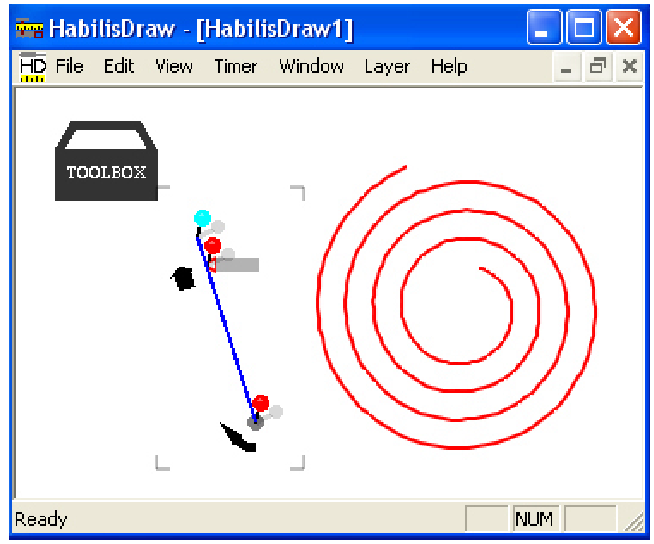

Figure 2: A power composite tool and its result.

2. HABILISDRAW 2.0

HabilisDraw is a prototype drawing application we have developed to investigate tool-based software environments. Comparable functionality can be seen in other drawing environments, in particular KidPad [1], but our focus on tools as distinct, first-class artifacts is novel. The current version of HabilisDraw[1] concentrates on the functionality of composite tools and power tools for producing drawings.

2.1 Original Tool Set.

The tool set in HabilisDraw 1.0 included pens, ink wells, pushpins, compasses, rulers, and lenses. This tool set, although not broad enough to be considered a complete drawing package, provided users with enough functionality to create engaging drawings, while allowing us to examine their use of the tools. Pens allow freehand drawing action. Ink wells can hold colors, pick up colors, and spill colors into objects. Pushpins act as handles on some objects for resizing. Compasses, as shown in Figure 1, allow users to create filled and unfilled circles and arcs. Rulers, also as given in Figure 1, act as a straight edge for pens and can be used to move and align objects. Lenses allow for the magnification of the page.

Most of these tools fit naturally into the framework given in Table 1. Pens are both effective and demarcating tools. Ink wells allow for the capture of information, namely ink color. Rulers allow for dynamic interaction with the pen, where the pen is constrained by the position of the ruler. Lenses are instruments, collecting information from the page and making it easier to see.

On top of these base tools we have built a set of power tools and composite tools. The power tool addition includes several new

1 See more images and a demonstration video of HabilisDraw 1.0 at http://www.csc.ncsu.edu/faculty/stamant/ibots.html.

tools, while the composite tool addition required the creation of more interactions. Figure 2 shows a power composite tool and its result. An explanation of this figure follows the discussion.

2.2 Power Tools.

There are currently six power tools in HabilisDraw 2.0: bars, movers, bar movers, Bezier bar movers, and rotators. These power tools share one key characteristic. Each contains an arrow that controls the intended action, with the arrow length and direction indicating the speed and direction of the action. Arrow placement and graphic design suggest the action's effect, as shown in Figure 1. A red arrow shows that the tool is in a neutral state that will result in no action. The arrow can be seen by selecting the tool with a left click of the mouse. At that point the arrow can be selected and dragged to the desired position.

Bars are created by bar pens. Bar pens function just like normal pens, but instead of leaving a mark they leave a rigid bar from the point of click to the point of release. Bars can rigidly constrain the spatial relationship between other tools; alternatively, they can be set such that their length will vary automatically. The arrow associated with each end of a bar can be pointed either toward the other end of the bar, for shortening, or away, for extension. In its inactive state the arrow turns red, indicating that the bar will not change in length when used.

To increase the flexibility of bar tools, we have added another basic tool to the HabilisDraw toolset, a Bezier curve tool, which allows the construction of Bezier curve bars. These curved bar can be reshaped by moving two points located along the bar.

A mover power tool simply moves across the canvas. It pushes any tools in its path in the way that a bulldozer pushes dirt. The arrow on a mover is positioned on the front, and can point in any direction; the length of the arrow controls the speed of the mover. Line and Bezier bar movers traverse bars and Bezier bars from one end to the other when attached to them. The arrows on these movers specify speed only. Rotators are spinning tools. The arrow curves along the edge of a rotator to indicate the direction of motion.

2.3 Composite Tools.

To create composite tools the user attaches existing tools to each other with pushpins. Pushpins have two states, pushed in and not pushed in (colored red and blue respectively.) A limited number of tool attachments are supported, but these can create a large set of composite tools. Any line can be attached to any mover or rotator. Any pen can be attached to any mover, rotator, or bar.

The motivation for composite tools is illustrated by the HabilisDraw 1.0 compass tool in Figure 1. In a conventional drawing environment, the user drags a bounding box over some area, and a circle is inscribed in that area. The compass tool gives a stronger visual suggestion of its function: once the compass is placed in some location, the user puts a pen inside its outer "handle" and simply draws. The structure of the compass constrains the drawing to circular motion around its center. After experimenting with other, simpler tools, we realized that just as in the physical world, the functionality of a compass can be met with a pushpin, straightedge (or even the equivalent of a string), and a pen, tools that already exist in HabilisDraw. The key is to extend the physics of tools so that they can be attached to one another and behave comparably to the way they do in the physical world.

The actions of the power tool transfer to the tool that it is attached to. A bar attached to a rotator is spun by the rotator. Anything attached to a mover moves along with it. The design of this interaction between tools turned out to be the hardest part of development. The tool serving as the anchor tool for chaining the actions must be decided on before performing the actions, and must be accurately chosen in order for the user's desired actions to occur. Once tools are attached and fashioned as desired a key combination will permanently group them within the program.

One final type of combination adds tools to the toolbox. The toolbox holds all of the existing tools in the system and any new composite tools that have been created. This allows users to customize HabilisDraw to specific applications.

The function of all the objects shown in Figure 2 should now be clear. A bar being spun by a rotator and growing in length allows for the creation of a spiral. The blue pushpin is not pushed in. The red pushpins are pushed in, attaching the pen to the bar and the bar to the rotator. The curved arrow controls the movement of the rotator; the straight arrow controls the growth of the bar. The movement of this tool, when activated in conjunction with a pen tool, creates a spiral; the tool is shown in its final state. The closed toolbox is where this composite tool can be stored for later use.

3. ACKNOWLEDGMENTS

Thanks to David B. Christian for his contributions to this project. This effort was supported by the National Science Foundation under award 0083281. The information in this paper does not necessarily reflect the position or policies of the U.S. government, and no official endorsement should be inferred.

4. REFERENCES

[1] B. B. Bederson, J. D. Hollan, A. Druin, J. Stewart, D. Rogers, and D. Proft, Local tools: an alternative to tool palettes, *Proceedings of UIST* (1996) pp. 169-170.

[2] E. Horvitz. Principles of Mixed-Initiative User Interfaces. *Proceedings of CHI* (1999), pp. 159-166.

[3] Maes, P. Agents that Reduce Work and Information Overload. *Communications of the ACM* 37(7): 31-40 (1994).

[4] St. Amant, R., and Horton, T. E., Characterizing tool use in an interactive drawing environments. *Proceedings of the Second International Symposium on Smart Graphics* (2002).

[5] St. Amant R., and Horton T. E., A tool-based interactive drawing environment. *Proceedings of CHI (Extended Abstracts)* (2002).

[6] Shneiderman, B. *Designing the user interface: strategies for effective human-computer interaction*, Addison-Wesley (1998).