

# **Embodied User Interfaces for *Really* Direct Manipulation**

**Kenneth P. Fishkin, Anuj Gujar, Beverly L. Harrison,  
Thomas P. Moran, Roy Want**

*Xerox Palo Alto Research Center*

A major event in the history of human-computer interaction (HCI) was the advent at Xerox PARC in the 1970s of the *Graphical User Interface* (GUI). The GUI was based on a bitmapped display, making the interface to the computer primarily a visual medium, along with a pointing device (typically a mouse), enabling the user to point to parts of the display. The GUI transformed HCI from communication in an arcane textual language to a visual and manipulative activity.

With the GUI, the virtual world inside the computer is portrayed graphically on the display. This graphical world is a metaphoric representation of artifacts in the office workplace, such as documents and folders. Icons represent these artifacts on the display, where they can be manipulated with the mouse, such as dragging a document icon into a folder icon to “file” it. An artifact can also be “opened up” so that its content can be seen and manipulated, such as by scrolling or by jumping between pages of a long document. The GUI paradigm, recognized as a breakthrough, was labeled “*direct manipulation*” by Shneiderman [6] in 1983.

It is interesting to note that while a whole sensory-motor world is created within the confines of the computer display, the physical computer itself—the workstation—has become an anonymous, invisible “box.” All the attention is on a disembodied display.

## Beyond the GUI

There are many directions in which HCI design is developing beyond the GUI. One direction—*virtual reality*—is to further enmesh the user within a high-quality, animated, 3-D world on the display. Pursuing this direction further, the display is worn as goggles, and the workstation “box” totally disappears!

We are interested in a quite different direction. Instead of making the box disappear, we want to rediscover and fully recognize that computation is embodied in physical devices that exist as elements in the physical world. This direction—*augmented reality*<sup>1</sup>—recognizes that the physical configuration of computational devices is a major determinant of their usability. There are several research efforts exploring augmented reality: Hiroshi Ishii’s “Tangible Bits” project at the MIT Media Lab [4], Jun Rekimoto’s Augmented Reality project at Sony Research Labs [5], George Fitzmaurice’s Graspable User Interface research at Alias|Wavefront and University of Toronto [2], as well as our own work [1,3]. We can also see this direction being pursued in the marketplace in portable computational “appliances,” such as “palm-sized” Personal Digital Assistants (PDAs, e.g., the Palm Pilot™) and the recent wave of electronic books or (E-books, e.g., the SoftBook™).

Several features of these new devices are noteworthy:

- The devices are *portable* and *graspable*: they must be held, touched, and carried to be used.
- They are designed not as general-purpose machines, but to support a limited set of *specific tasks*.
- The work materials are *contained inside* the devices, and thus the devices *embody the tasks* they are designed for.
- The device casings are physically designed to make these tasks *easy and natural* to do.
- The devices are *metaphorically related* to similar non-computational artifacts.

For example, consider the 3Com Palm Pilot™. It is light, small (pocket-sized), and graspable in one hand, and has a pen to be used by the other hand. It is designed to support four specific tasks (calendars, to-do lists, address lists, and brief notetaking). It is designed to be like a paper personal organizer. The user’s calendar data is in the Pilot, and thus the Pilot *is* their calendar. The task of setting an appointment is the task of writing it on the Pilot, which is designed to make it easy to do.

---

<sup>1</sup> Mark Weiser has promoted the similar concept of *ubiquitous computing* [7], where computation is distributed throughout the physical world. He has also termed this approach “embodied virtuality.”

## Embodied User Interfaces

The physical interaction with such devices is still quite limited. Like a traditional workstation, interaction with these devices is through a pointing device (a pen rather than a mouse) on a display, plus a few physical buttons. But compare this to a paper artifact, such as a notebook. We not only write on it, but we flip, thumb, bend, and crease its pages. We have highly developed dexterity, skills, and practices with such artifacts, none of which are brought to bear on computational devices.

So, why can't users manipulate devices in a variety of ways—squeeze, shake, flick, tilt—as an integral part of using them? That is what our research is investigating: Can we design natural manipulations? Can we implement hardware to robustly sense and interpret the manipulations? When are such manipulations appropriate?

We want to take user interface design a step further by more tightly integrating the physical body of the device with the virtual contents inside and the graphical display of the content. By treating the body of the device as part of the user interface—an *embodied* user interface—we can go beyond the simulated manipulation of a GUI and allow the user to *really* directly manipulate an integrated physical-virtual device.

There has been much recent work in the research community on a variety of techniques embracing some of these principles. For example, researchers have explored interfaces in which a user scrolls through a menu by tilting the display, zooms text by pushing/pulling the display, or explores maps by moving talismans representing various buildings about the display surface (see [3] for a list of references). For our work, we were interested in exploring the potential of embodied user interface techniques by

- focusing on common electronic tasks for which a strongly analogous physical task already exists,
- designing and implementing examples of the techniques,
- testing them on users to obtain experimental feedback, and
- developing a general framework for designing embodied user interfaces.

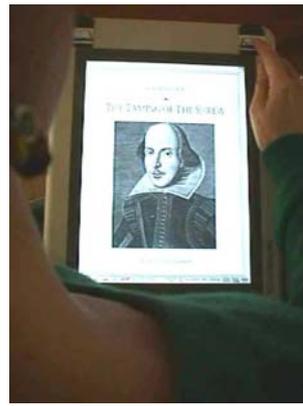
In this paper we illustrate embodied user interface techniques by considering paper document handling tasks. People have developed a highly refined set of physical techniques to perform these document tasks. With the recent interest E-books, we thought it would be useful to investigate how exploit these familiar techniques in an embodied user interface. We discuss here three different tasks on three different devices. The first two tasks concern traversal through sequences of pages or notecards. The third task concerns annotating documents.

## Example Task 1: “Turning” Pages in a Document

Given a handheld device that holds multi-page documents, we want the user to be able to navigate through them by simply “turning pages.” The particular device we’ve chosen to explore is a portable computer display, a Mutoh-12™, which is an off-the-shelf display with desirable properties for E-books (page-sized screen, XGA resolution, pen input, etc).



(A)



(B)

**Figure 1. Turning pages with a real book (A) and with an enhanced pen computer (B).**

### Design

Our approach is to design a new *device-embodied task* in relation to some familiar *analog task* that users are skilled at. In this case, the obvious analog task is turning physical pages in a paper book. To embody this task in a computational device, the device must naturally represent as many of the critical properties of the analog task as possible. Documents must be represented as a sequence of pages with only one or two pages being displayed at a time. Our challenge, then, is to allow the user to change the displayed pages on the device in a manner similar to a paper book.

Our approach to designing embodied user interfaces is that the device-embodied task follow a *Physical-Effects Principle*: the *virtual* effect of a physical manipulation in an embodied user interface should be compatible with the *physical* effect of that manipulation in the analog task. For example, in the real world, users can turn to the next page with a right-to-left flick on the upper right corner of a page, and turn to the previous page with a left-to-right flick on the upper left corner, as shown in Figure 1A. Our design, therefore, used the same manipulations: a new virtual page is displayed after a physical “flick” is sensed on the device, as shown in Figure 1B.

### Implementation

Hardware to support these flick manipulations detects finger pressure in the upper-left and upper-right corners and the direction of finger movement. We decided to put pressure sensors on the frame of the device, rather than make the display pressure-sensitive. We

did this because the sensors were easy to retrofit. They are independent of particular software applications. They do not require any visual indicators on the screen, and gestures on the frame will not be confused with other gestural commands that might be given on the screen.

### **User Reactions**

We tested and iterated on our initial page turning design several times. In the first implementation, the sensors required too much pressure. Users expected much lighter flick gestures from their experiences with paper books. In order to match these user expectations, we altered our implementation to detect lighter strokes.

Initially, we designed the sensors so that a right-to-left “flick” on the upper right corner would display the next page and a left-to-right “flick” on the upper left corner would display the previous page. However, the latter gesture was not only inconvenient, but also the user's arm temporarily obscured the display. We redesigned the sensors so that either upper corner of the frame could be used for either forward or backward page turning, as indicated by the direction of the “flick” gesture. We signaled the location for page turning gestures by open-book graphics on the upper corners of the frame, as shown in Figure 1B. Users had no problem in discovering the manipulation needed for “previous page” once they had tried the “next page” manipulation. Users relied on their understanding of the analog task to guide exploration of the device-embodied interface.

Early versions of the system provided visual feedback by displaying the new page immediately upon detecting the flick. However, this did not provide the user with any sense of which direction they are moving in the page sequence. Furthermore, users could easily miss seeing the page change, since it occurred so rapidly. Thus, we animated the pages flipping in the appropriate direction on the display, in accordance with the physical-effects principle.<sup>2</sup>

In addition to visual feedback, we also explored audio feedback in one implementation, where we provided optional audio page-turning sounds during the animation. In general, the design of aesthetically pleasing sounds is extremely difficult. In practice, users often do not wish to have their computers making noises that might disrupt others. We found that most users turned this feature off.

## **Example Task 2: Traversing a Sequential List**

The next task involves a user traversing a sequence of items. Traversal is more rapid than page turning, because the user is usually skimming through the sequence to find a particular item. The specific example we chose was a contact list, and the embodied device we chose was a Palm Pilot™.

### **Design**

For the analog real-world task, we consider how people use a Rolodex™, a popular physical device for managing contact lists. A Rolodex™ holds a sequence of index cards

---

<sup>2</sup> In our most recent implementation, we extended the page-turning flick gesture to support a “thumbing” gesture, where the user can press on the upper corner to cause the pages to rapidly flip in succession. This combined two functions (rapid skimming and single-page turning) in related gestures. For this to work well, the animation rate increases and the timing had to be carefully adjusted to avoid misinterpreting single page turns. Overall, users seemed to learn these gestures quickly and claimed they were very intuitive.

on a wheel, as shown in Figure 2A. Each card holds data on one contact, and the cards form a circular list. Usually only one card near the top of the wheel is visible. As the wheel is turned, different cards move into this visible position. The user turns the wheel by a knob on the side. This manipulation allows the user to scroll through the list in either direction, to control the rate of scrolling, and to stop scrolling when the desired card is in view.



(A)



(B)



(C)

**Figure 2. Navigating through a list, with a real Rolodex™ (A) and with an enhanced Pilot™, front (B) and side (C).**

To design a device-embodied task on the Pilot, we created a software application that displayed the contact list as a stacked array of tabbed cards, as shown in Figure 2B. The main design issue was how to enable the user to traverse the list. We decided to make

tilting the Pilot cause the list of cards to scroll in the direction of the tilt. The response of the device is, according to the physical-effects principle, that the cards fall in the downward direction of the tilt. The analogy with the Rolodex™ is that turning the knob causes a “circular tilt” of the wheel. Just as turning the knob makes scrolling faster, greater tilting makes the list on the display scroll faster. Tilting is relative to a “neutral angle” at which the user normally holds the Pilot to view it without any scrolling.

A second design issue was how to stop the scrolling action. In an early design we had the user stop the scrolling by tilting the device back to the neutral angle. However, we found that this was not an accurate enough manipulation for users. Considering the Rolodex™ knob again, we see that there is a subtle braking manipulation to stop the turn. We decided that a reasonable braking manipulation on the Pilot would be to squeeze the device to stop the scrolling cards, in analogy to the user having to squeeze the Rolodex knob to stop it.

The final design issue is a general one for this kind of user interface: How does the device distinguish *intentional* from *inadvertent* manipulations? A Pilot is often tilted as the user carries it around, sets it down, or uses it in any way that violates our assumption about the neutral angle. We decided to require an explicit manipulation by the user (again a squeeze) to signal intentionality. The squeeze manipulation was convenient, and itself is not an inadvertent action. Thus, the user squeezes to start tilt “mode,” tilts to cause the scrolling action, then squeezes again to stop the scrolling. To further suggest squeezability, we put foam padding around the Pilot casing (Figure 2C)..

### **Implementation**

We mounted a commercially available tilt-sensor on the back of the case of the Pilot, with the sensor axis parallel to the plane of the display. We found that we only needed to distinguish between 16 different degrees of tilt.

To support squeezing, we attached pressure sensors along both sides of the Pilot in positions aligned with the user’s fingers and thumbs (Figure 2C). To differentiate intentional squeezing from simply holding the device, we tested ten users to derive a threshold value.

We put circuitry on the back of the Pilot to monitor the sensor values and convert the analog signals into digital samples, which were then transmitted along a serial cable monitored by the application.

### **User Reactions**

We tested and iterated the design many times. One issue was determining the range of angles for the tilt operation and the value of the “neutral angle.” We determined the initial neutral angle by in-laboratory testing, which turned out to be about 45 degrees. The range of tilt angles was partly based on just noticeable differences, both in terms of user-discernable tilt angles and in terms of user-discernable scrolling speeds. The range of perceptible tilt is an important factor when setting and assigning values for the tilt manipulation’s parameters. At present the tilt angles map to 6 different scrolling rates.

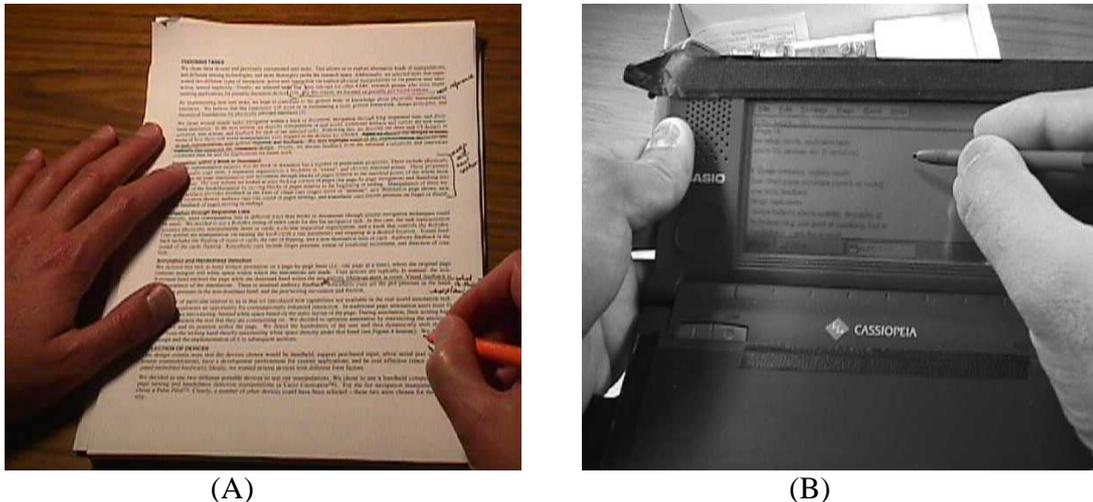
We found that our slowest scrolling speed was set too fast, as users tended to overshoot the target item. We learned that it is necessary to fine-tune continuous manipulations that control rate and/or direction. We are investigating this issue further to determine how much variation amongst users affects their ability to precisely control list

manipulation. Some alternative manipulations may be useful, with one type of manipulation for coarsely specified actions (e.g., fast scrolling), followed by a second manipulation for finely specified actions (e.g., deliberate scrolling).

Finally, as a consequence of using tilt to control list navigation, display visibility was an issue. In particular, we avoided use of extreme angles of tilt, since the Pilot display was not readable at these angles. Different devices and/or displays have different viewing angle restrictions, which must be taken into account if the display plays a central role.

### Example Task 3: Annotating a Document

The final task involves assisting users in making handwritten annotations on document pages. We want to keep the analogy to paper-based annotation on a computer with a pen input device, while exploiting the potential of the computer to provide *additional* capabilities. For this design, we chose a third device, the hand-held Casio Cassiopeia™, because of its commonality and its low cost. (Also, we wanted to test our interfaces on as many different devices as possible.)



**Figure 3. Annotating a document on paper (A) and on an enhanced handheld computer (B).**

#### Design

The analog task is annotating a page of a paper document with a pen, as shown in Figure 3A. The page usually contains margins and other white space where the annotations can be made. User's actions are typically bimanual: the non-dominant hand anchors the page while the dominant hand writes the annotations. The user must fit annotations into the existing limited space. Also, the writing hand often obstructs the content of the page.

In designing the device-embodied task, we saw an opportunity to make the annotation task easier by dynamically shifting the displayed text to increase space for annotations. The optimal place for annotation is on the side of the page where the user's writing hand is, as shown in Figure 3B. Thus we wanted to unobtrusively detect the user's handedness.

We observed that users of these kind of handheld devices typically grip the device by the left or right edge with the non-dominant hand and use the dominant hand to write on the display. We did not want the user to have to make an explicit signal to communicate handedness, so we decided to sense handgrip. When only one side of the device is being gripped and it is in an application that wants to be “handedness-aware,” the device shifts the display contents.

### **Implementation**

To detect user handedness, we again used pressure sensors. We attached pressure-sensing pads to the back of the device casing on the left and right sides, where users hold it. When a sensor detects pressure, the document display is immediately shifted towards that side, allowing the user more space on the other side, where the free hand can write with a pen.

Since this shifting happens automatically, with no explicit user action, we were worried that users could inadvertently cause undesirable effects by resting the device on a desk or on their lap. However, because the sensors were placed behind the lid of the device, these situations did not cause the system to be fooled.

### **User Reactions**

Passively sensed handedness detection worked amazingly well. It detected correctly, and users did not need to alter their natural usage of the device. All users remarked on the “magical” nature of this feature. Since no explicit commands were employed, users seemed amazed that the device recognized handedness. They were unable to tell how this was accomplished without us explaining it. This suggests that passive manipulations can be powerful, and that they can greatly impact a user’s interaction experience when well integrated with the device. We believe that the success of this feature is partly due to in-laboratory pre-testing. We tested 15 users to fine-tune the placement of the pressure pads to best accommodate variations in hand size and hand grip location.

Another explanation for the strong positive reactions was the augmentation of real-world capabilities. By optimizing annotation space, we created a function that does not exist in the corresponding analog situation. This illustrates an opportunity for computationally augmented task representations that positively enhance the real world analogy. However, to create natural enhancements (as opposed to unexpected or non-intuitive ones), the system must confidently “know” what the user wants to do. This matching of system response to user intention is crucial. In this case, annotation support worked well because our assumptions accurately predicted user goals.

## **Conclusions**

Our test users generally found the manipulations “intuitive”, “cool”, and “pretty obvious in terms of what was going on.” Some users needed quick demonstrations to understand that their manipulations would actually be interpreted. They had little or no exposure to embodied user interfaces and therefore often did not expect interaction with the device to be understood. Conveying the basic paradigm will be necessary just as users needed to understand the conceptual foundation for GUI interfaces. Once users understood the basic paradigm, they immediately began to explore the range of interaction. Just as GUI users try to find out what’s “clickable”, our users tried a variety

of manipulations on the prototypes to explore the space of detectable manipulations. For example, to turn pages they tried long and short strokes, fast and slow strokes, light and hard strokes, and starting the stroke at different points on the device surface.

This new interaction paradigm usually involves putting more physical sensors on a device, a significant design decision. This decision seems to involve a tradeoff between a richer interactivity and cost. We believe the decision to utilize an embodied user interface makes sense under particular circumstances: when the device is a focused information appliance that involves functions that are particularly useful, frequently needed, inherent in tasks supported by the appliance, and amenable to physical actions from a familiar analog tasks. Commercial example of this rationale can be found in some of the new E-books that are using rocker switches or pressure strips to support page turning.

There are a host of design issues to explore in embodied user interfaces. What is a good framework for design that incorporates these new techniques? How do we evaluate which manipulations are best for which tasks? How are embodied user interfaces best integrated with other input techniques, such as audio? How are embodied user interfaces best employed to handle complicated command sequences? Some of our recent work has begun to address these questions by laying out a design framework specifying the design space and organizing the issues involved [1].

We hope that these examples have shown the potential for *Embodied User Interfaces*. By treating the device itself as a “first class citizen” in user interface design, and widening the range of physical manipulations with which users interact with the device, we feel that more natural and effective user interactions are possible.

## Acknowledgements

This work emerged as one aspect of a Xerox research effort, coordinated by Polle Zellwegger, devoted to exploring new document reading devices. We thank all those involved from Fuji Xerox Palo Alto Lab, Xerox PARC, and Xerox dpiX for their contributions.

## References

1. Fishkin, K. P., Moran, T. P., and Harrison, B. L.. Embodied User Interfaces: Towards Invisible User Interfaces. *Proceedings of EHCI'98* (Heraklion, Greece), Sep. 1998. In Press.
2. Fitzmaurice, G., Ishii, H., and Buxton, W. A. S. Laying the Foundations for Graspable User Interfaces. *Proceedings of CHI'95*, pp. 422-449.
3. Harrison, B. L., Fishkin, K. P., Gujar, A., Mochon, C., and Want, R. Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. *Proceedings of CHI'98*, Los Angeles, CA, April 18-23, pp. 17-24.
4. Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms. *Proceedings of CHI'97*, pp. 234-241.
5. Rekimoto, J. Tilting Operations for Small Screen Interfaces. *Proceedings of UIST '96*, pp.167-168.
6. Schneiderman, B. The Future of Interactive Systems and the Emergence of Direct Manipulation. *Behavior and Information Technology*, 1, 1982, 237-256.
7. Weiser, M. The Computer for the 21st Century. *Scientific America*, 265, 3 (Mar. 1991), 94-104.