Chapter 3:

ALTERNATIVE PERSPECTIVES

Extending Our Understanding of the Relationships Among Devices

In the previous chapter, the grain at which we looked at input devices was fairly coarse, especially if our orientation is the user and usage, and not the technology. If we want to probe deeper, characterizing devices as "mice", "tablets" or "joysticks" is not adequate. While useful, they are not detailed enough to provide us with the understanding that will enable us to make significant improvements in our interface designs.

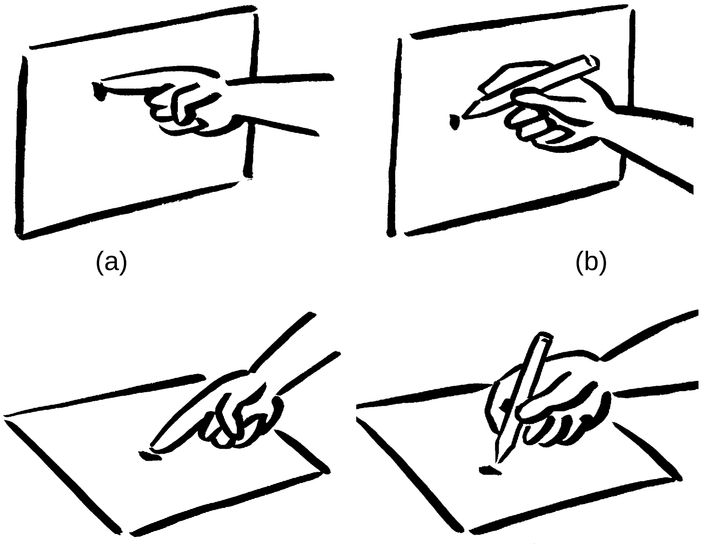
The design space of input devices is complex. In order to achieve a reasonable grasp of it, we have to refine the grain of our analysis to something far finer than has hitherto been the case. In the sections which follow, we explore some of the approaches to carving up this space in ways meaningful to the designer.

If design is choice, then developing a more refined taxonomy will improve the range of choice. And, if the dimensions of the resultant taxonomy are appropriate, the model that emerges will afford better choices.

As a start, let us take an example. It illustrates that - even at the top level - the dominant mouse, joystick, trackball ... categorization is not the only way to carve up the "pie."

Figure 1 shows a caricature of four generic devices:

- a touch screen
- a light pen
- a touch tablet
- a tablet with a stylus.



(a) (b)

(c) (d)

Figure 1: Analogy and relationships among different devices
The devices characterized in this figure possess some important properties that help us better understand input technologies in context. For example, they hold the relationships A:B=C:D (A is to B what C is to D), and A:C=B:D. (Drawing by Bart Kip.)

Each device illustrated may use a very distinct technology, and in many texts each would be treated in a distinct section. However, the example lends itself to other interpretations. Alternative criteria may form the basis for categorizing the devices. To begin with, let us look at two which - significantly - cut in two orthogonal directions. One dimension is *screen vs tablet*. The other is *touch vs stylus*.

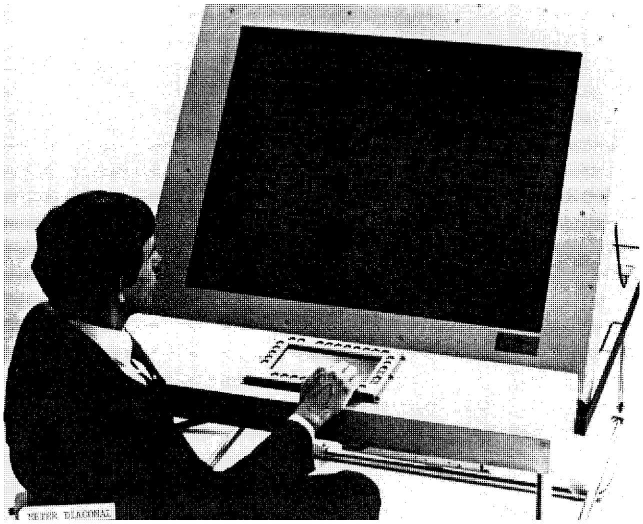


Figure 2: A touch screen functioning as a touch tablet.
In this example, a flat panel display with a touch screen is mounted horizontally and functioning as a tablet. While this input device is a display, it is not the primary display. It is used for control purposes only. The distinction between touch screens and tablets is blurred. Orientation of the control surface and the presence of another display seem to be more significant dimensions for characterizing the system.

In recognizing such relationships, we stand to gain some valuable insights. For example, from the screen *vs tablet* relationship we see:

- that the touch screen and light pen share the advantages of "directness," that is, the person interacts directly with the displayed object;
- that the touch screen and light pen also share many of the same disadvantages, such as arm fatigue (when the display is mounted vertically) and the potential of obscuring the target with the pointer.
- Along the touch *vs* stylus dimension we see:
- the potential differences in resolution of pointing between stylus and finger;
- the absence of the stylus "tip switch" and its resulting signals when pointing with the finger. What is the effect of this on the repertoire of transactions that can be supported?

From the former emerges the relationship that the touch screen is to the light pen what the touch tablet is to the tablet with stylus (A:B=C:D). According to the second, we see that the touch screen is to the touch tablet what the light pen is to the tablet with stylus (A:C=B:D).

In studying the example, we see that there is yet another way to categorize the technologies, namely *orientation of the control surface:* flat *vs* vertical. This is another interpretation of the A:B=C:D relationship. Different questions emerge once we recognize this relationship. Consider the case where a display with a touch screen and light pen are mounted horizontally rather than vertically.

- We see that the property of arm fatigue mentioned above is not an attribute on the directness dimension, but on that of orientation;
- Assuming a flat panel display and equal resolution, we see that the notion of orientation and direct *vs* indirect are far more meaningful than the notion of "tablet." In fact, at this stage it becomes difficult to understand just what a tablet is.

The GRidPad personal computer, shown in Figure 3, is a further example of how it is less the technology than its situated context, or usage, that should serve as the basis for any taxonomy that we might develop.

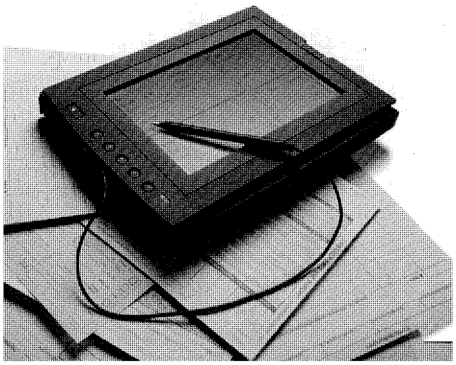


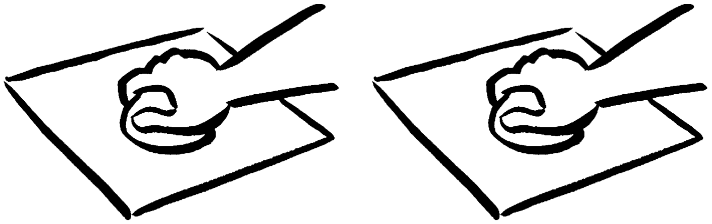
Figure 3: The GRidPad Personal Computers

Like a lightpen and tablet, the GRidPad uses a stylus for input, directly on the display surface. However, the technology used is different so that the approach to input is better characterized by its style (character recognition) rather than by the technology used.

Like the tablet and light pen, the GRidPad uses a stylus. And, like a light pen, one uses the stylus directly on the display surface. While sharing many of the properties typically encountered with lightpens, the technology used is quite different. The dominant aspect of the system's approach to input is not the technology, but the style: character recognition.

Another relationship that is interesting to explore is that between a mouse and a tablet. The grain of analysis will be about the same as the previous examples. Let the tablet be controlled by a puck (rather than a stylus) and the mouse use a mouse pad. Also, assume that the mouse and the puck have the same size, shape, and buttons and where the tablet and the mouse pad are the same size. This is illustrated in Figure 4.

Where does the "tabletness" and "mouseness" of the two devices lie? Certainly not in their physical properties. To the eye, the two may be indistinguishable. Along at least one dimension (the hand-held transducer), the tablet with puck has more in common with the mouse than it does with the tablet with stylus seen in the previous figure. Along this same dimension, the tablet with stylus has more in common with the light pen (especially when its display is mounted horizontally) than the tablet with puck.



(a) Digitizing tablet puck

(b) Mouse with Mouse Pad

Figure 4: Comparison of Tablet and Puck and Mouse with Pad

Even from this coarse representation it can be seen that the tablet and puck have a great deal in common with the mouse and pad. It takes little to imagine that the tablet puck and mouse could look exactly the same, and that the tablet base could be exactly the same size as the mouse pad. Where, then, is the essence of "mouseness" or "tabletness"then? When claims are made for one of the devices, when are they based on their shared attributes and when on their distinct ones? Is it always clear? (Drawing by Bart Kip.)

Probing deeper, consider the case of a tablet which has a template, or menu mounted on its surface, such as shown in Figure 5. What does this say about directness? The user is interacting directly with the display of the controls (the menu item on the tablet). Compared to the earlier examples of the touch screen and light pen, is this direct or indirect? Having tried to answer that question, what if the tablet with template is replaced by a flat panel touch display, with the templates drawn by software? On the one hand, this is just a horizontally mounted touch screen. On the other hand, if it is used as a secondary control display, it is fundamentally different than a touch screen mounted on the primary display itself. Hence, another significant dimension emerges: *congruence of the display and control surfaces*.

In this dimension, a tablet and touch screen with templates have more in common than a conventional touch screen (touch control on primary display) and a secondary touch screen with templates used for control.

These examples have led to the recognition that there are a number of different ways of classifying devices, besides the traditional mouse, tablet, joystick, trackball, ... categorization:

- *Directness:* are the control and display surface the same?
- *Pointing tool:* finger, stylus, puck, ...

- *Orientation:* horizontal, vertical, ...
- *Congruence of display and control surface:* for example, is the touch screen on the primary display, or a secondary display used as type of touch tablet.

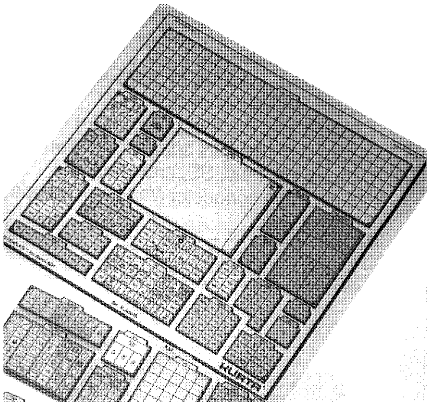


Figure 5: Tablet with Template on Surface

Tablets can have templates mounted on their surface delimiting menu items of special functions, as illustrated in this example. The control and display surface (at least for the menu items) are the same. Consequently, a tablet used in this way starts to assume the property of directness seen in the light pen and touch screen. (Figure, Kurta Corp.)

These are just the relationships that have emerged from the selected examples. Like in mining, if you find some nuggets, chances are, there are more yet to be discovered. To find those that are not exposed to the surface, we need to develop better techniques and methodologies of prospecting. Likewise with input, and that is the path which we will follow in the coming chapters.

Input/Output Mappings

The previous section used physical properties of the tools to help establish some relationships among devices. In this section, we look at properties which have to do with the mapping between the property sensed by the transducer and the feedback that occurs in the system from that action. We will see that this is one dimension of comparison that helps, for example, establish a meaningful difference between the puck with tablet and mouse with pad shown in Figure 4.

We will approach these mappings by talking about the relationship between property sensed and feedback. To set a context, our first example considers the case of trying to position a tracking cross on a screen.

With a tablet in "absolute Mode," for example, the current position of the the screen's tracking cross directly corresponds to the current position of tablet puck or stylus. There is a direct position-to-position mapping.

With a "floating" joystick, such as on the *Skedoodle* toy, we find a similar position-to-position mapping.

With the self-returning joyswitches found on most arcade video games, however, the mapping is position-to-motion. Moving the joystick away from its centre position causes the tracking symbol to move in the same direction. The amount that the joyswitch is displaced from it's "home" centre position has no effect on the speed of the tracking symbol's motion. Control is of direction, only.

There are, however, video games and computer systems that use *spring-loaded self returning* joysticks which do control both direction of motion and speed: the further you push the stick in a particular direction, the faster the tracking symbol moves in that same direction.

In contrast to all of the previous examples, mice and trackballs are interesting in that their position is irrelevant. With them, the mapping is motion-to-motion. While with a mouse, for example, this might often feel like position-to-position, it is significantly different. This is most obviously seen when one tries to trace a drawing or pick a mouse up from the left side of the mouse pad, and place it down on the right. With a tablet puck, the tracking cross would move. With the mouse, it remains stationary.

We can summarize these and other mappings in table form for comparison. This is done in Table 1.

Property Sensed	Feedback	Examples
Position	Position	Light pen, touch screen, tablet puck and stylus
Position	Motion (direction only)	Joyswitch
Position	Motion (direction & rate)	Self-returning rate joystick
Motion	Potion (direction & rate)	Mouse, trackball
Force	Motion (direction & rate)	Isometric joystick

Table 1: Mappings between property sensed and feedback for positioning task
The table shows various mappings between control and feedback, with examples, for a task involving positioning a tracking symbol on the screen.

Our first example used positioning. In the real world, we generally find devices used in many other types of task. By going through the same type of exercise for another task, we will lay the foundation for additional insights.

Let us consider the case where we scale a value or object, or transform it, such as by rotation. Here, rather than specify position, we are specifying a magnitude (of size, value, rotation, etc.). In this case, a linear potentiometer might be used to specify a position-to-magnitude mapping. Similarly, we could map the position of a tablet stylus to the magnitude of the value.

With a joyswitch, however, we would implement a position-to-growth mapping. As was the case with positioning, the position would only determine the direction of growth, not the speed. With a self-returning rate joystick, however, both direction and rate of growth could be controlled.

These and other cases are summarized with examples in Table 2.

Property Sensed	Feedback	Examples
Position	Magnitude	Linear potentiometer
Position	Growth (direction only)	Joyswitch
Position	Growth (direction & rate)	Self-returning rate joystick
Motion	Growth (direction & rate)	Mouse, trackball
Force	Growth (direction & rate)	Isometric joystick

Table 2: Mappings between property sensed and feedback for scaling or rotation task

These mappings afford the distillation of some observations concerning system design and human performance. The most important point, which is virtually axiomatic, is:

Within the bounds of equal demands on motor/sensory skills, X->X mappings will virtually always result in the least cognitive load.

For example, all things being equal, motion->motion or position->position mappings will result in lower cognitive load than force->motion, or position->growth mappings. The underlying reason for this is the directness of the compatibility between stimulus and response.

The "catch phrase" in the above is the qualifier regarding motor/sensory skills. All transducers are not created equal. Some are inherently harder to operate than others. This may be because of poor design, or the fact that humans just happen to be not very good at the type of control captured by the transducer.

There may be, therefore, a trade-off between cognitive *vs* motor/sensory load. The obvious design objective is to minimize both, and thereby avoid the trade-off. Much of what follows in latter sections has to do with investigating the human performance parameters that affect this trade-off.

Time vs Space Multiplexed Control

There are two generic approaches to the design of input controls:

- *Space Multiplexing:* in which each function to be controlled has a dedicated transducer.
- *Time Multiplexing:* in which one device is used to control different functions at different points in time.

Space multiplexed systems derive their name from the fact that there are several controllers spread over space. The layout of the controls in an aircraft's cockpit, as illustrated in Figure 6 is one example. An automobile is another, perhaps more familiar, example: each of the brake, clutch, throttle, steering wheel, and gear shift is a distinct transducer dedicated to controlling a single specific task.

In contrast, the mouse on a Macintosh computer is a good example of time multiplexing. Here, the same transducer performs one task one moment, and another task the next. It controls functions as diverse as menu selection, navigation using the scroll gadgets, pointing, and activating "buttons" such as those on the Desk Accessory calculator. Time multiplexing gets its name because the single physical transducer performs different functions over time.

The Macintosh serves to illustrate one other point: there need not be an either/or choice between time and space multiplexing. For example, the split between functionality assigned to the mouse *vs* the keyboard is space multiplexed. The Macintosh, therefore, is be seen to employ a *hybrid* approach to its control structure.

Space Multiplexing

With space multiplexed architectures, there is minimal confusion over what function is performed by any device. The problem is that as the number of functions increases, so does the number of input devices. The large number of controls that may result can increase cost, work place "real estate" and training time. Aircraft cockpit design is a good example where the initial approach was almost completely space multiplexed. To reduce problems induced by the increased functionality of flight systems, more recent designs have incorporated a high degree of time multiplexing in both controls and displays.

One very positive aspect of space multiplexing is that the design can ensure that each device has a fixed physical location. When this is true, the trained operator can, through *motor memory*, memorize device locations and access controls without visual distraction. The standard size and layout of keyboards, for example, takes advantage of this property and thereby affords the ability to *touch type*. The fixed position of an automobile's gear shift lever permits the skilled driver to exercise analogous *eyes-free access* to the control *and its associated functionality*. (This last point is important. For example, a joystick might be located in a fixed location which a skilled operator may commit to motor memory. However, if its functionality is time multiplexed, its operation will most likely still require visual attention.)

Figure 6: Space Multiplexed Controls: an Aircraft Cockpit
Note how the controls are distributed in space. Aircraft present an extreme example of space multiplexing. The proliferation of controls can cause confusion. Hence, there are trends in modern cockpit design to time multiplex some displays and controls. (Photo by author.)

Time Multiplexing
Time multiplexed systems are inherently highly modal. This is ironic, since it precisely highly time multiplexed systems like the Macintosh that claim to be "modeless." xxxxxx more to come.......

In design, it is important to address the time/space multiplexing trade-off. This is a subject that is woefully neglected in the HCI literature. There are a couple of points that we can make now, and which we will expand upon further in later sections of this book.

Let's assume that our system will adopt a hybrid approach. The first question that arises, therefore, is where do we draw the line? How many devices should we have, what should they be, and how should tasks be assigned to them?

Unfinished- to add: *display often space mpx while input time mpx. give impression, provides feedback to avoid confusion. on other hand, number of mouse buttons issue is largely result of different functions assigned at different times w.o. adequate feedback. Here, case of time mpx device (namely mouse button) and no feedback. Argument is usually formulated as one of degree of space multiplexing (how many buttons), when errors are largely due to inappropriate time multiplexing of the individual buttons. Since feedback difficult in such cases, approach to button problem is training, and to assist/accelerate that, consistency - change function of individual buttons (time mpx) with great care. Mouse buttons can be considered easy to reach function keys. With careful design, they can be used effectively with minimal load.*

Admit, multi button approach can hurt performance on "10 minute rule." But don't forget the "easy to reach" factor. May be buying into a save 10 minutes of training once on first learning the system, vs a lifetime of penalty in performance. It is not a question of load or no load. It is a matter of when you pay: cash up front, or infinite installments.

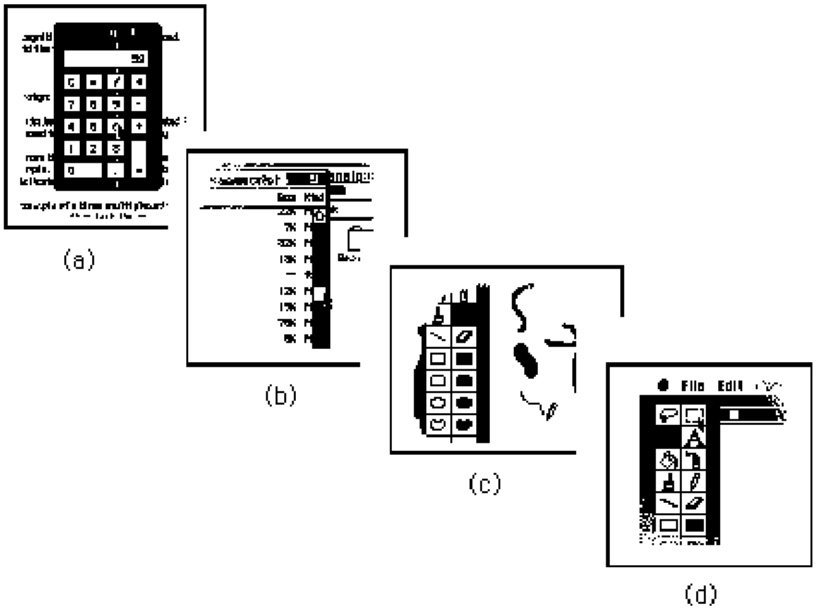


Figure 7: Time Multiplexing Input on the Apple Macintosh

The figure shows how a single physical device (the mouse) assumes the function of different virtual devices over time. The figure shows the mouse as a virtual numerical keypad (the calculator), navigation control (scroll bar), drawing tool and menu selector. One benefit is that economy of physical transducers with the lower cost and reduced footprint that results. However, with the approach, only one of the tasks can be performed at a time. They are mutually exclusive because the all use the same transducer.

Notes:

- Add Rob Jacob's stuff on Perceptual Structure (InterCHI, also expanded in ToChi 1(1))