Sketching User Experiences

getting the design right and the right design
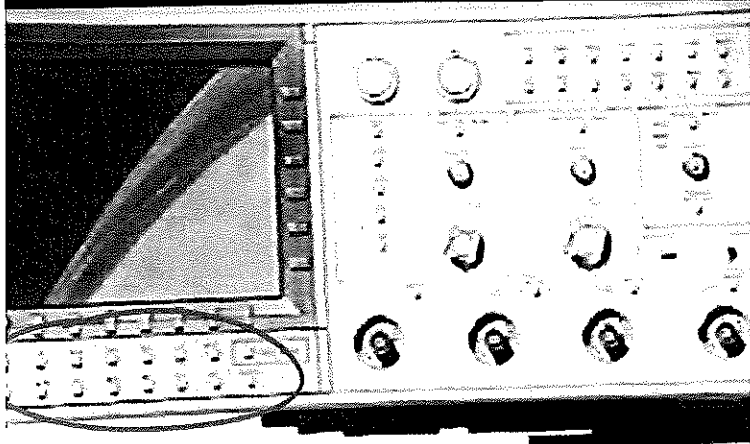
Bill Buxton

Focal
Press

M K

Figure 141: Paper Sketches of Oscilloscope Front Panel

Two frames from a videotape made during usability testing of oscilloscope front panel concepts. The usability of the numerical keypad is being tested. In each version, I have circled the location of the keypad.
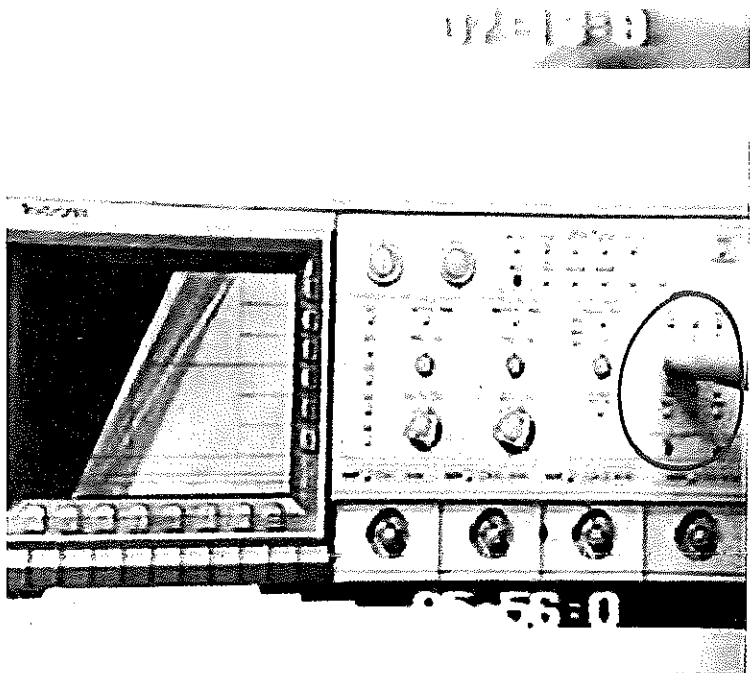
Image: Tektronix Corp.

Figure 142: Interacting with Paper Front Panel

This is a frame from the same video. In this image, and the second one in Figure 141, notice how the hand gesture used by the test user is exactly what would be used with the physical control represented. This is a subtle but important point that illustrates the power of the affordances of the controls, even on paper.

Image: Tektronix Corp.

Inte

... success
— Mick F

ə Front Panel
⸴ usability testing of
bility of the numeri-
n, I have circled the

Interacting with Paper

... success every time implies that one's objectives are not challenging enough.
— Mick Fowler

nel
image, and the sec-
gesture used by the
the physical control
ioint that illustrates
, even on paper.

We now move back to the world where sketches are interactive and they can be experienced first-hand. Curiously enough, to start things off, we are going to go back to where we started: with traditional sketches on paper.

Example: Scoping Out Tektronix

Sometime in the 1980s I was invited to do some consulting for Tektronix, a company in Portland, Oregon. While I was there, I had the opportunity to speak with some of the people who designed their test instruments (core products of the company at the time). Earlier, I had published a paper on iterative design (Buxton & Sniderman 1980). So in typical fashion, a little too self-important, I asked them, "So, do you do iterative design as part of your process for developing the front panels for your oscilloscopes?" Knowing me well enough by then, they promptly answered, "Yes, of course. We do exactly one iteration per device, and then we ship it."

Okay, I got what I deserved. But after explaining the facts of life to me, such as the costs (in time, labour, and dollars) of building prototypes of various panel designs, they showed me what they *did* do. It was a revelation.

This was, I believe, my first exposure to what some people call "paper prototyping," but which I simply refer to as *paper interfaces* (more on why, later). What they did was have one of the designers make a quick drawing of the proposed front panel, and then bring in expert users to test them. The tests were videotaped for later analysis and comparison. What the sessions consisted of was a member of the design team explaining a particular scenario to the user, and then having the user perform the appropriate task using the sketch as a proxy for the planned product, all the while talking aloud explaining their actions and intentions, as well as asking any questions. Figures 141 and 142 are frames showing two different panel designs from one such video.

What is interesting in viewing the tapes is that the scenarios are imaginary. There is nothing displayed or moving on the paper oscilloscope's screen, and during the tests, nothing changes in the drawings. As long as the appropriate scenarios and users are selected, this turns out not to be a problem (but it may, of course, restrict what tests or users could be employed).

In the case of the two screens shown in Figure 141, what is being tested is the location and layout of the numerical keypad. Due to cost and space constraints, the mechanical engineering team had proposed locating the keypad underneath the display, as illustrated
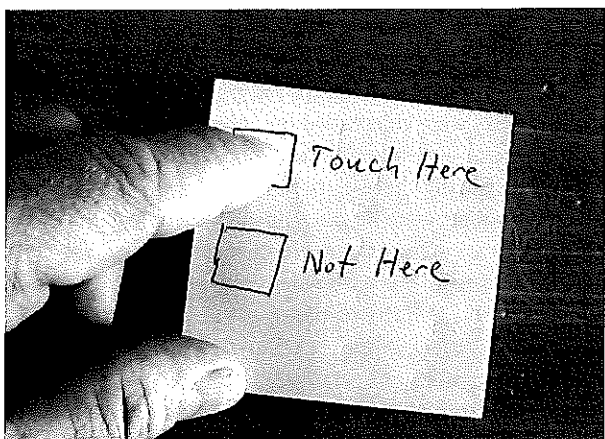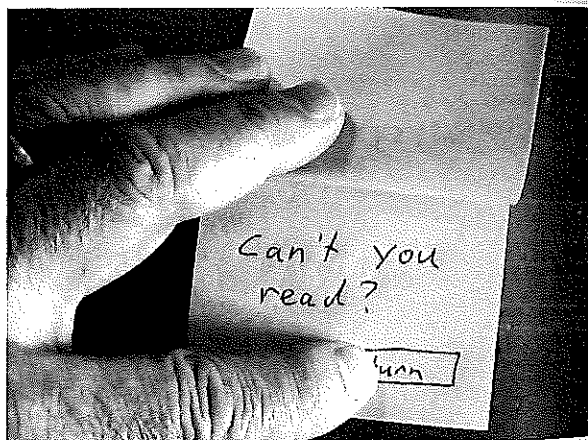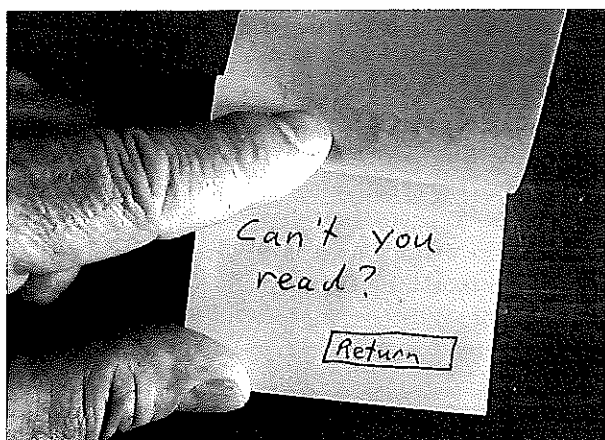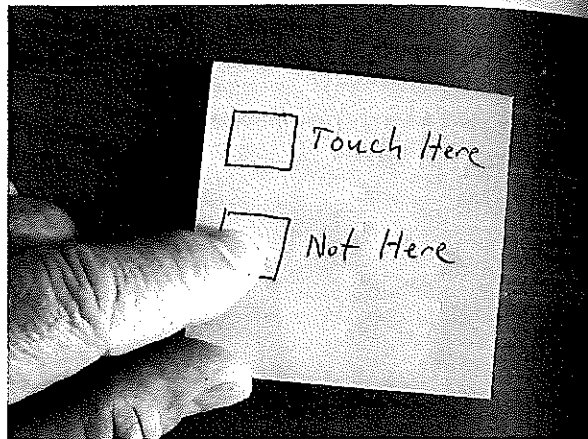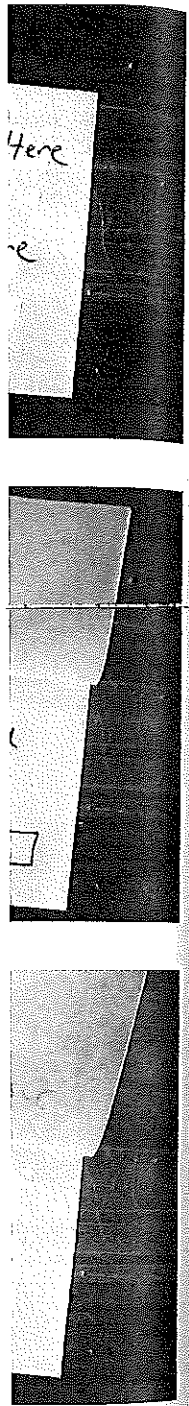
Figure 143: A Simple Finger Exercise
One can create and experience an interactive paper interface in two minutes with nothing more than Post-it notes and a pen. Push a button to go to a particular page. Push the wrong button and return to the first page.

in the top image. Using that sketch, expert users were asked to perform various tasks that involved the keyboard. However, as can be heard on the video, they frequently asked questions like, "Which keypad?" or "Now what keypad do you mean?" or "Where is the keypad?" This prompted the designers to change the position and layout of the keypad, as per the second image. When this revised design was subjected to the same tests, the questions about the keypad simply disappeared.

In retrospect, the problems with the initial version of the keypad may seem obvious. But of course, hindsight is 20-20. Their root is in our previous experience with telephones and pocket calculators. These have acclimatized us to expect numerical keypads to be arranged in 3x4 arrays, as in the second, as opposed to the 8x2 arrangement seen in the first.

By using paper, the design flaw was caught early, and an alternative design was quick to generate and test. Because of the speed of the technique, and the compelling nature of the results, the revised design was able to be incorporated into the product before the mechanical tooling needed to be ordered.

Finally, there is one other observation that I want to make. Look at Figure 142 as well as the second image in Figure 141. In each I chose a frame from the video that includes the user's hand. What struck me when I first saw these clips was how powerful the affordances of the drawn controls are. There can be no confusion on the part of the operator that this is just a piece of paper. Yet the hand posture over the rotary and push-button controls is exactly what one would encounter with the actual physical device. To me, this speaks volumes as to the power of paper to evoke real-world behaviours.

Interactive Paper Interfaces

With the Tektronix oscilloscope example, we saw an example of interaction with a static sketch. Now we introduce paper interfaces that are *dynamic*. That is, what you see in the sketch changes depending on one's interactions with the drawn interface. Like the Tektronix example, the technique requires someone who assumes the role of the target *user*, and someone from the design team, the *facilitator*, to guide that user's experience.

The basis of the technique is very simple. If the user interacts with something on the sketch, that should cause a change in what is seen. For example, if the user pushes a button, the facilitator makes the appropriate change happen. This might be accomplished by something as simple as replacing one sketch with another. A simple two-minute exercise, such as that shown in Figure 143, lets you get a feel for this. Make a simple interface with a pen and a Post-it note pad, and be your own facilitator. Then try it on someone else while you facilitate.

The fact that my example is trivial is a good reason to replicate it, rather than not to. Because it is virtually contentless, you can focus on the technique. And, because it is so simple, the contrast in your (cognitive) understanding before and your (experiential) understanding after will be all the more marked.
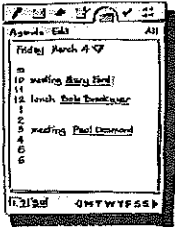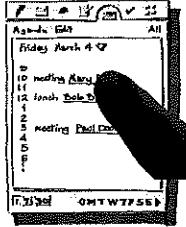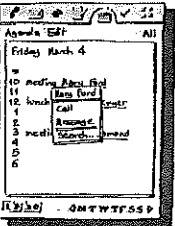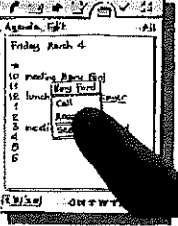
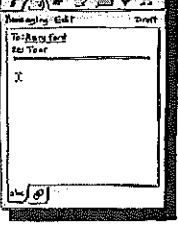|	Facilitator	User
(sketch: Agenda Edit, Friday March 4, meeting Mary Ford, lunch Bob Breakwear, meeting Paul Demond)	(Start: Sketch 1.a in front of user.) The sketch in front of you shows the screen of your PDA. I want you to send a message to your 10:00 am appointment. For this exercise, to do anything, just touch what you think is appropriate on the screen, and tell me what you are doing or thinking as you go along.	
(sketch with finger touching Mary)		Okay. I assume that you want me to send a message to Mary Ford, since she is my 10:00 am appointment. So I will touch her name.
(sketch with Call/Message/Arrange menu)	(Facilitator replaces sketch 1.a with 2.b)	Now I see a menu that lets me either call her or message her.
(sketch with finger touching menu)		So, what I will now do is touch "message" on the menu.
(sketch: Messaging Edit, Draft, To:Mary Ford)	(Facilitator replaces sketch 2.b with 4.a)	Okay. I now see a screen that lets me send a message to Mary Ford. What now?

Table 2: Example of Interacting with Dynamic Sketched Paper Interface

The user is guided through a transaction with the agenda originally seen in Figure 99, (pages 284-285). Each time the user's action would result in a change in a real system, the facilitator gives the user an appropriate replacement sketch with which to continue.

that you want
ssage to Mary
s my 10:00 am
I will touch her

u that lets me
message her.

w do is touch
e menu.

a screen that
essage to Mary
)

ction would result in a

Back to Our Agenda

Now let's apply this kind of technique to a less trivial example. To do so, we will repurpose the drawings of agenda screens that we saw in our discussion of storyboards, as illustrated in Table 2.

In this scenario, the facilitator starts by showing users an initial screen image, 1.a (with the screen map at the bottom cut off). The user then is asked to send a message to his 10:00 A.M. appointment. Each time the user takes an action the facilitator makes sure that the current screen image is replaced by the new appropriate one.

Read the cells in the table left to right, top to bottom. In the left column of each row, you see what the subject sees, as well as what he is pointing at and selecting with his finger. In the middle column you read what the facilitator says, and in the right column, the remarks by the user. The session is hypothetical. I made it up just for the example, but it is representative of what you will encounter in the field, nevertheless. Try it for yourself and you will see.

In the example, notice how a very simple change in representation (adding some text and a finger, and cropping out the screen-flow map) enabled the images of Figure 99 to communicate different aspects of the interface design. Literacy in such techniques and representations is one of the most important tools in the repertoire of the experience designer—knowing what to use when, with whom, and how.

The example shown in Table 2 is still verging on the trivial. Again with sketching that can be a good thing. Unlike grade school, doing anything beyond the minimum required to achieve your objective is a waste, not something that gets you a gold star. Fast and simple is what lets us afford to explore multiple concepts. But at some point we will need to go deeper. One approach to doing so is to take our sketches electric. That is, use a program like *Flash, PowerPoint* or *Director*, for example, to make a computer-based interactive sketch. In this case, Ron Bird built an interactive program in Director to capture the concept. A screen snap of this is seen in Figure 144.

In this implementation, Ron did a couple of clever things. First, he did not try and implement the interface in a way that the user could use it. It would simply have taken far too much work to bring either the design or the implementation to a point where one could let users loose on it. All that he wanted to do at this stage is capture some of the basic concepts. Instead, the interaction involves the user selecting aspects of the user interface that he or she wants to see. The options are listed in the text column on the right side of the figure. When the screen snap was taken, the user had selected the second item, and the cursor can be seen there.

The trick to doing this was to distinguish the real cursor (what the user selected the right-hand list item with), from the representation of the cursor driving the sketched interface (represented by the sketch of the stylus).

What I also like about this example is how Ron preserved the sketch-like quality of the graphical rendering that we saw earlier in his paper sketches. This is no coincidence. He did the images on paper and then scanned them in to the computer. The effect reinforces the fact that these are not finished ideas.

Later, as the testing converges on a design that they have more commitment to, it can make sense to progress to the next step and make a more interactive model that can be
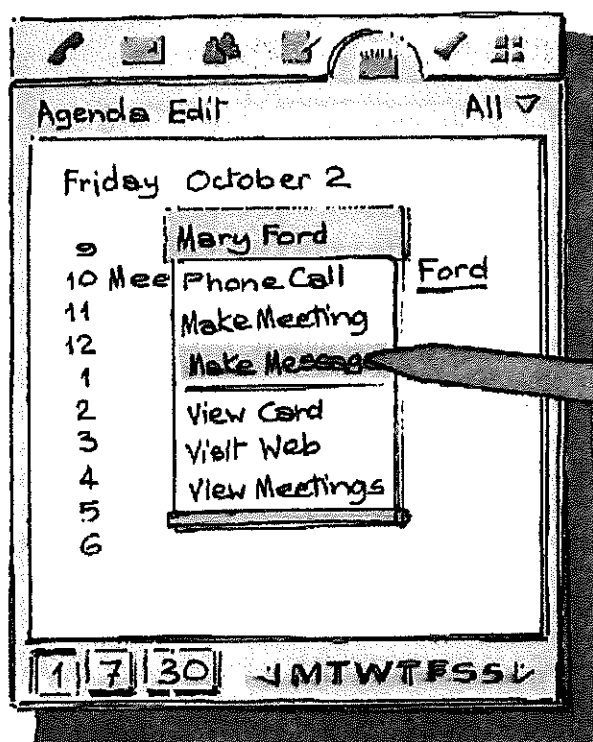
Figure 144: A Computer-based Sketch of the Agenda Interface

This is a screen snap of an interactive computer-based sketch of the agenda seen in the previous example. In this case, the user can explore aspects of the interface listed on the right (the user-controlled cursor can be seen having selected "What a link can do." This invokes an animation, where the user's interactions are represented by the graphical stylus. It is important to note that Ron has preserved the hand-drawn sketch-like character of the interface. It is clearly not done. This is just a probe.

Image: Ron Bird

operated directly by the user. But doing so brings us closer to the world of prototyping than sketching. It has its place, but it is not our primary focus now.

So instead, we are going to back-track and explore an alternative approach—one that lets us get interactive much sooner, much faster, and much cheaper—by sticking with paper.

The approach that we are going to take builds on the paper techniques used in the agenda interface just described. Using various materials that can be found at any business supply store, we can make interfaces that are far more interactive than the paper ones we have seen thus far.

To illustrate this, we are going to work through an example. But if you want to dive a bit deeper into these, a good, short introduction can be found in Rettig (1994). Try it first. Then if you really need or want to go into way more detail, see the book by Snyder (2003) which is excellent. Just don't be intimidated by any of this. Kindergarten provided you with most of the technical skills that you will need.

So here is what we are going to do next:

- Get a better taste of paper-based techniques, just so you have a bit better sense of the overall flavour.
- Talk about what is wrong or missing in most discussions about so-called paper prototypes.
- Explore how these techniques fit into the larger picture of interaction design, especially in terms of mind-set and methodology.
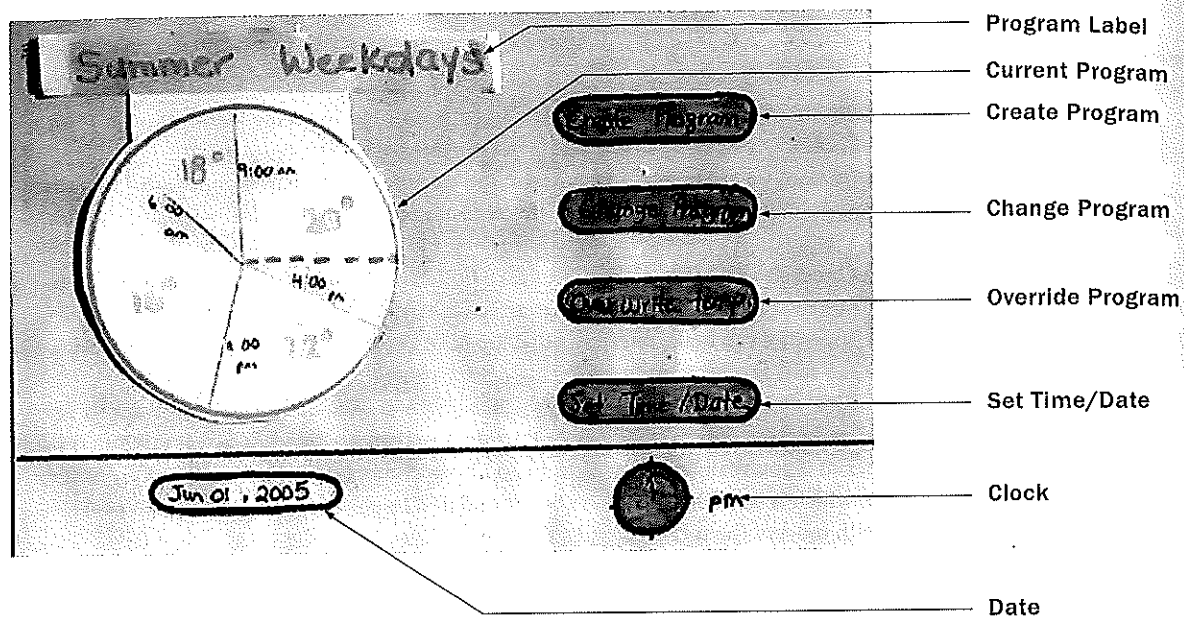
Program Label

Current Program

Create Program

Change Program

Override Program

Set Time/Date

Clock

Date

Figure 145: Paper Interface to a Programmable Climate Control System
The basic interface is made up of buttons and circular dials, and displays. The
concept is that the user would interact directly on the screen by means of a
touch screen.

n Label

Program

Program

Program

e Program

e/Date

Keeping Cool—A Home Climate Controller

The example that I am going to use was done by a graduate student at the University of Toronto, Maryam Tohidi, working with Ron Baecker, Abi Sellen, and myself (Tohidi, Buxton, Baecker & Sellen, 2006a).

Maryam's brief was to do an interactive paper sketch of a home climate control system. The system had to enable the user to do things like program the temperature for different times of day and for different days, such as weekends and weekdays, summer versus winter, and so on. One version of the basic home display is shown in Figure 145, along with labels for its main components.

The exercise is intended to simulate a system where one touches things on the display to exercise control. The design language is based around a circular dial. It can represent different things, such as the duration of the day, or the four seasons of the year. In Figure 145, the dial represents the 24 hours of the day. The "slices of the pie" represent distinct intervals of the day, each with its own temperature, as indicated by the text on the slice.

Some of the steps in creating a new program are shown in Figure 146. This illustrates some of the techniques that the facilitator can use to make the paper display dynamic, in the sense that it can respond to the user's interactions with it.

Another technique is shown in Figure 147. In this case, instead of moving tape, or using replaceable overlays, the facilitator is erasing what is written on the interface, and then writing in new information with a dry marker. The interface is paper underneath. It just happens to be covered with acetate, on which it is easy to write and erase, using markers. Yet another trick from grade school.

I have focused on the mechanics of manipulating the paper interface, not on how you might use it. In that regard, there are many options, including the following:

- If the intent is to quickly explore a concept, or show it to your colleagues, the designer might play the role of both user and facilitator, and use the paper interface to walk/talk through the design. In such cases, the designer may well revise the interface on the spot, based on comments, and then immediately start again with the new version. Such a walk-throughs might be captured on videotape for future reference, or to communicate the ideas to those not physically present.

- For informal testing, or quick probes, the designer might play the role of facilitator and work through the interface with someone representative of the intended end user or customer. Often recording such sessions on video is useful, since it is hard to capture the user's comments, or notice all of the subtleties of their interaction with the interface, if you are trying to "operate" the paper at the same time. As in the previous case, the design may be revised on-the-fly, based on comments that it generates.

- Then, there is their use in usability testing. Here the objective is more to uncover errors and determine usability than to come up with new design concepts. The purpose is to get the agreed upon design right, rather than determine, from among alternatives, what is the right design. The interface generally is tested with several users, and for results to be valid, the interface cannot be changed from user to user.
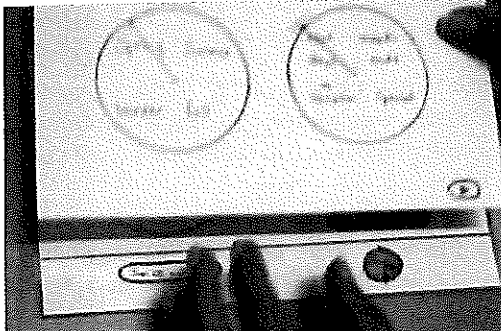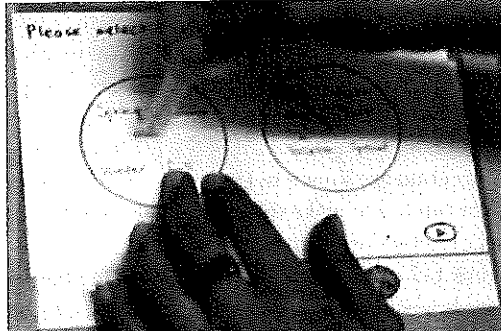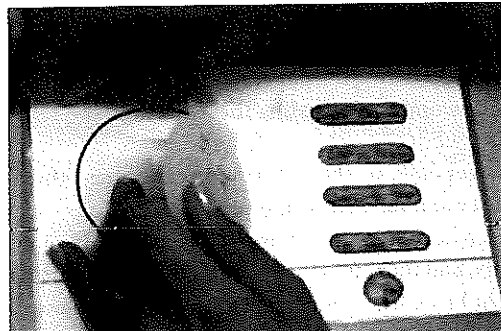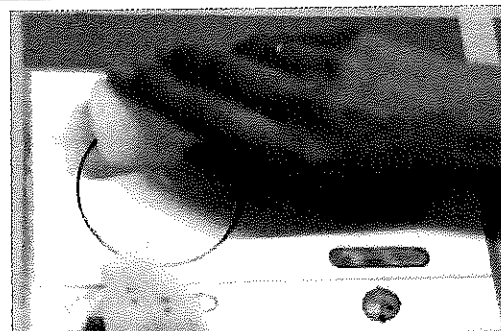
379

	If the user pushes the Create Program button, the main part of the display is replaced by two dials. The left one shows the four seasons, the right one four options: "Week Day", "Weekend", On Vacation", and "Special". The user selects the season and type of day by touching the appropriate "slice" of the display, or dragging the red dial indicator.
	The indicator is actually a piece of transparent tape that is stuck to the dial. The glue is like that on a Post-It. That is, it can be easily lifted up and stuck down in a new position. That is what the facilitator is doing in this image: moving the indicator to reflect the season chosen by the user.
	When the new program is set, the facilitator returns to the original screen, shown in Figure 145, and updates the Program Label.
	The "face" of the dial is also replaced with one that reflects the new program.

Figure 146: Creating a New Program

The third case typically is used later in the design cycle when there is already a fair degree of commitment to a particular design. The size of the investment at this point is higher, consequently, the process is more formal than in the previous two cases. Hence, besides the users, there is a larger team of people required.

As commonly practiced (e.g., Rettig 1994), these include the *facilitator*, who talks with the user and works them through the exercise. There is a separate person, the *computer*, whose sole job is to manipulate the paper interface in response to the user's actions and according the conventions of the design and the test procedure. There is a *videographer* (or two), responsible for capturing the session—a two camera shoot is generally best, since you can then get both a close-up of the interactions with the interface, and a wide shot showing the context and things like general body language. Then there is one or more observers who take notes during the session. Finally, there is often a greeter, who manages logistics, and essentially stage-manages the whole thing.

Hmmmm. That sounds like a lot of people and a pretty heavy-weight process. Despite the sketch-like qualities of the paper interface, it doesn't sound much like sketching when we test it according to attributes such as quick, cheap, timely, disposable, and lots of them.

In fact it isn't. I want to push on this point a little harder.

Appearances Can Be Deceiving

Hopefully you remember my earlier assertion:

> **Sketches are not prototypes.**

As I have mentioned, the use of paper-based interfaces generally is referred to as *paper prototyping*. This is a term that I have been very careful to avoid during our discussion. Likewise, I have avoided using the terms high-fidelity prototype and low-fidelity prototype (Rudd, Stern & Isensee 1996). The reason in both cases is rooted in the earlier assertion. A prototype is a prototype, regardless of the technology that is used to implement it, and regardless of its fidelity relative to the actual product. Prototypes are not sketches and the term low-fidelity prototype is definitely not a synonym for my use of the term sketch.

So here is my cautionary note:

> **Just because something looks like a sketch doesn't mean that it is a sketch.**

Here we can reap the benefit of having been explicit about sketch attributes. Rendering style was just one of them, and on its own, is certainly not sufficient to make something qualify. As the heading of this section says, appearance can be deceiving. This shouldn't come as a surprise. After all, that is the whole premise of the *Wizard of Oz* technique. Sometimes the deception due to appearance can work in our favour, and sometimes not.

In the case of paper interfaces, I think that the appearance can too easily lull us into the impression that we are sketching and designing, when in fact, we are using inexpensive prototypes to do usability engineering. This is certainly what Snyder's book is about, rather than design. This is not a bad thing. Both are important. We should just be conscious of the difference and its significance.
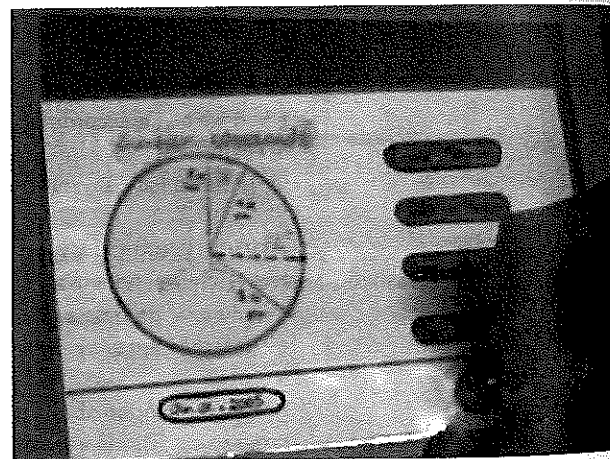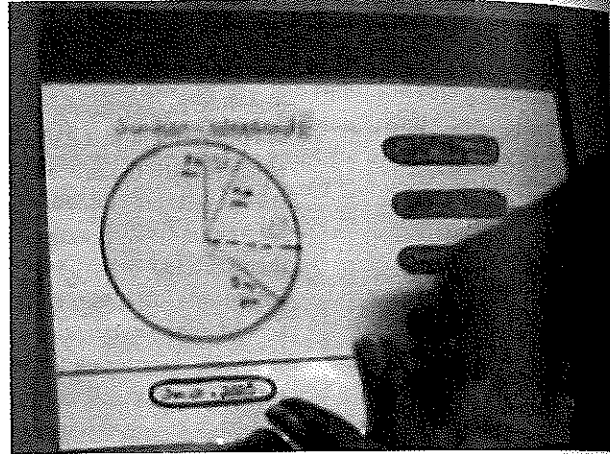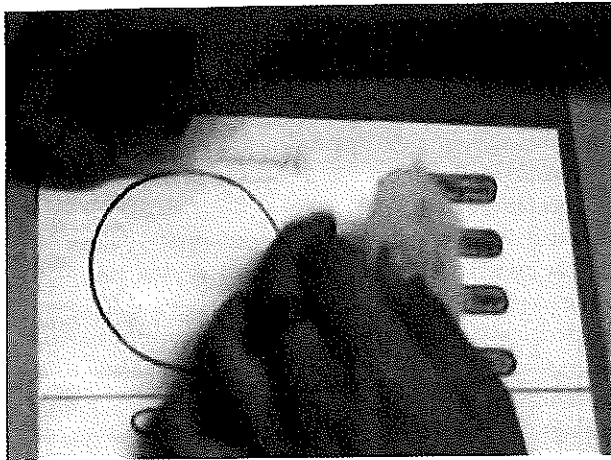
Figure 147: Changing the Display by Erasure and Writing

By covering the paper with plastic, one can write on it with a dry marker, and have what is written easily erased with a cloth when the information needs to be changed. Sometimes this is easier than having a stack of premade objects to stick down.

Sketching is not just *what* you use, but *how, when, where,* and *why* you use it. *That* is what determines if you are sketching versus prototyping. In the previous section, for example, I outlined three possible scenarios for using the paper climate controller. I would suggest that the first two are good candidates for qualifying as sketching, and the last one is definitely not. It is far more consistent with prototyping.

The importance of all this really came home to me during the early stages of researching this book. I was reading an article entitled, *Does the Fidelity of Software Prototypes Affect the Perception of Usability?* (Wiklund, Thurrott & Dumas 1992). In the authors' words:

> The objective of this study was to investigate whether the aesthetic refinement of a software prototype is related to subjects' ratings of the usability of the prototype.

This was a pretty good study, and one that is relevant to our work. After all, if I am going to advocate using rough sketches to test and explore design concepts, it would be nice to know if there is any scientific evidence that the results have any relevance to the real thing. So let me explain a bit of what they did. I'm going to do so for three reasons: first, because it is an interesting study; second, because of what I said earlier about learning about each other's traditions; third, because it feeds into our discussion of sketching and ideation versus prototyping, usability, and engineering.

Wiklund et al. took an existing software package—one that combined the functions of a dictionary, calculator, and thesaurus—and retroactively made four prototypes of it. Each of the four prototypes varied in the fidelity with which it visually reflected the actual product. Two of the four are illustrated in Figure 148. The one with the lowest fidelity was done using line-art, the next one using half-tone, the next in gray scale, and the highest fidelity one was done in colour.

They were all interactive, and each was tested and rated for three things: ease of learning and use, forgiveness of errors, and aesthetics. The key finding was:

> ... the aesthetic quality of prototypes within the range we varied did not bias users for or against the prototype's perceived usability. (Wiklund, Thurrott & Dumas 1992)

So all of this was really interesting to me, and was relevant to my work. But it was *nothing* compared to what Wiklund and his coauthors said, almost in passing, in the discussion of their results. Here is what they wrote:

> In studies such as this one, we have found subjects reluctant to be critical of designs when they are asked to assign a rating to the design. In our usability tests, we see the same phenomenon even when we encourage subjects to be critical. We speculate that the test subjects feel that giving a low rating to a product gives the impression that they are "negative" people, that the ratings reflect negatively on their ability to use computer-based technology, that some of the blame for a product's poor performance falls on them, or that they don't want to hurt the feelings of the person conducting the test. (Wiklund, Thurrott & Dumas 1992)

It means what it means.

For thos
ments, o
carefully
the gen
tions in
style. Tl
have be
they su
finding:
the gol

I say
wish th
across
that thi
by this

The
author
reasor
me, I
somec
in my
book's
interp
the sc

For those who are not accustomed to interpreting the findings of psychology experiments, on page 383 I have underlined the qualification that Wiklund, Thurrott & Dumas carefully and properly included in their article. It is always dangerous to extrapolate to the general case from a specific experiment's results. For example, despite the variations in fidelity, none of the prototypes in this study were rendered in a sketch-like style. They were all "drafted", so the experiment does not tell us if the results would have been the same if they had been drawn freehand. Yet, they are relevant, since they suggest that we may be pointing in the right direction. Consequently, read these findings as, "This looks like a good spot to go prospecting," not as "Wow, here is where the gold is, let's open a mine."

I say this as if what I just warned you about is a trap that only amateurs fall into. I wish that that was so. In the course of doing the research for this book, I have come across countless examples of authors—all of whom should know better—making claims that this paper shows this or that about "low fidelity prototypes," when what they mean by this term bears little or no relationship to what Wiklund et al. actually studied.

The experiment means what it means, and that is all. The frequency with which authors stretch the interpretation of other's results to suit their own purpose is the reason that I always read such things with a skeptical mind. When it is important to me, I go back and read the original reference before acting on the interpretation of someone else. That is also why I don't cite papers second-hand, and why I possess in my personal library, and have gone through individually, virtually every entry in this book's bibliography. It would have saved a lot of time and money to trust other people's interpretations. But I can't and won't. The savings are not sufficient compensation for the scholarship that would be lost.
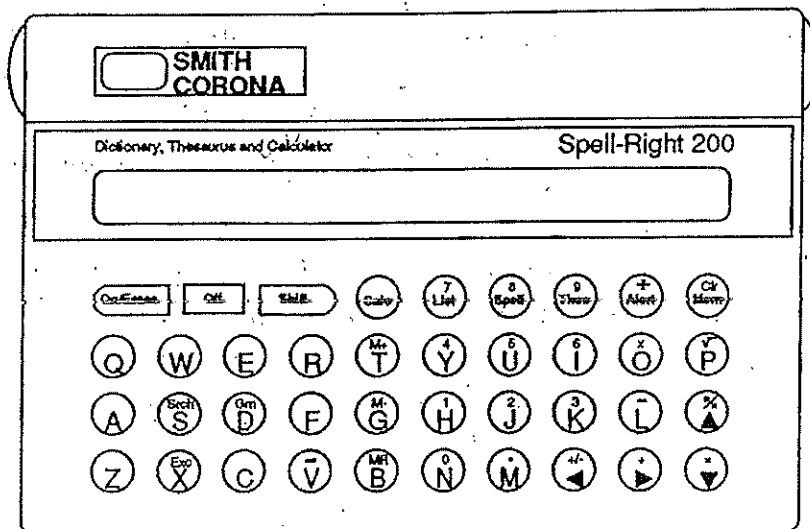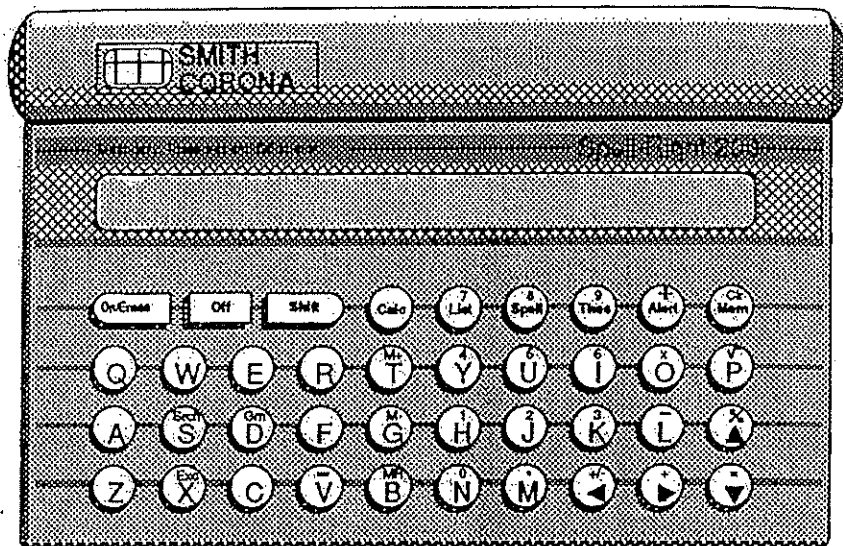
Figure 148: Line Art and Half-Tone Prototypes

These are two of the low-fidelity interfaces for the digital calendar used in the study by Wiklund, Thurrott & Dumas (1992). Contrast their formal drafted graphical style with the freehand renderings of Ron Bird's agenda sketches in Figures 99 and 144.

This really struck home with me. First, it rang a very strong note of familiarity. Second, it brought up an important issue that I probably would have not otherwise mentioned.

My immediate reaction on reading this was that Wiklund *et al's* encountered the kind of behaviour that they reported *because the subjects only saw one version of the dictionary.*

Just think about how the subjects' behaviour might change if they were exposed to three or four different versions of the prototype, rather than to just one. That could put them in a very different position. Their comments and evaluations would be grounded on a broader base of relevant experience. When they see just one design, judgments are absolute, but on a scale relative to what index? Seeing multiple alternatives can help set the scale so that it is relative to what is possible, rather than just to what the user has seen in the past.

Now we are no longer talking about Wiklund *et al.'s* study, but a fundamental difference between design and usability engineering, at least as it is currently practiced.

From my perspective, usability engineering follows a process that can be characterized by the drawing that I have made in Figure 149. One is on a single, already established trajectory (represented by the red arrow). An iterative process (represented by the black conic spiral) is followed that eventually converges on the completed product. The trajectory is established by the basic design, which is already done, and the iterations are there in order to test and refine its implementation, and fine-tune the design.

The key thing to note is that the process is one of incremental improvement where the result of each build-test-evaluate cycle is an improved version of the previous cycle. There are not major changes in the design trajectory, or any backtracking, unless some fundamental flaw is found.

In contrast, the purpose of design is to *establish* the trajectory that we saw in Figure 149. Hence it both *precedes* usability engineering, and is *complementary* to it. Furthermore, it is distinct, because of the very different nature of the process, as I have tried to capture in Figure 150. Rather than a converging spiral, the branching structure shown in the figure is how I think of it.

In this case, the purpose of the red line is to emphasize how many branches were being explored at that given time in the process. Hence, what I am describing is completely consistent with the definition that I gave earlier:

Design is choice.

The various branches shown in Figure 150 are intended to represent the various alternatives that were explored in the process of arriving at the end design, which is represented by the branch at the extreme right. This is where the red arrow in Figure 149 begins.

If I was asked to try and say what these two figures are intended to represent in the fewest number of words, I think that I would say this:

The role of design is to find the best design.
The role of usability engineering is to help make that design the best.
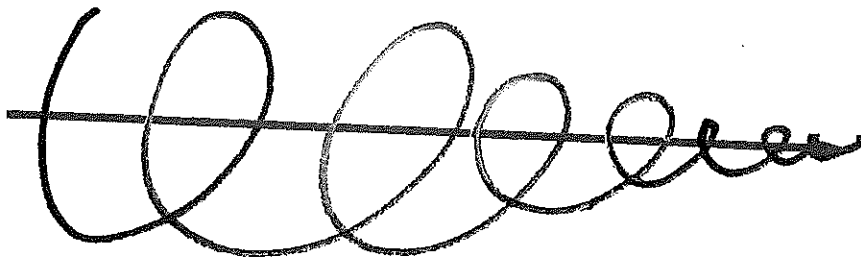
Figure 149: Prototyping as Iterative Incremental Refinement
In engineering, prototyping is like a spiral closing in along a single trajectory. Each prototype is a refinement of the previous one, and takes you one step closer to the final product. Iterative prototyping is a form of incremental refinement and validation, rather than a technique of exploration.
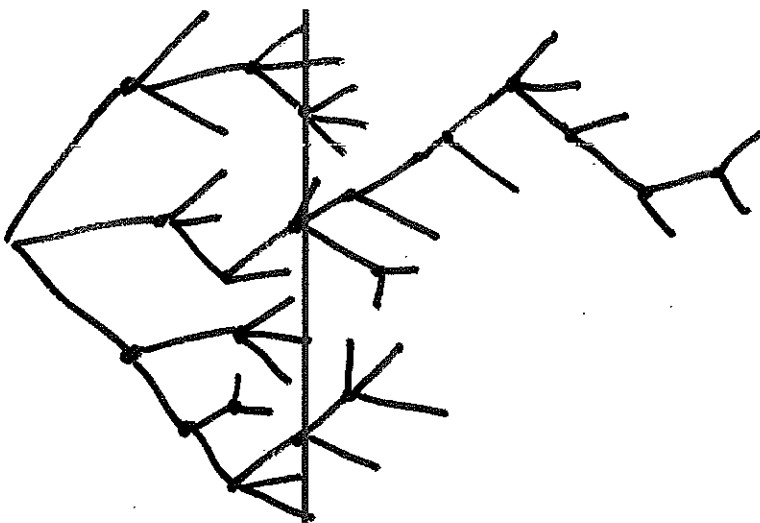

Figure 150: Design as Branching Exploration and Comparison
Design is about exploring and comparing the relative merits of alternatives. There is not just one path, and at any given time and for any given question, there may be numerous different alternatives being considered, only one of which will eventually find itself in the product.

Another way of saying this is:

> The role of design is to get the right design.
> The role of usability engineering is to get the design right.

In many ways, these are just variations of the distinction that we talked about earlier between problem setting and problem solving (Schön 1983)

So now let me go back to the study by Wiklund et al. and tell you what thoughts were triggered by the passage from their discussion.

The first was the following hypothesis:

> If we asked users to rate three distinctly different design solutions to performing a particular task, and then showed the design with the lowest rating alone to a comparable population of users, the users who saw just the poorest design would rank it significantly higher than those that saw the other two designs as well.

So why do I care if this is true? More to the point, what difference might it make to you?

- The comments by Wicklund et al. suggest what a lot of us already suspected: that we may not be able to trust the ratings that subjects give us about our designs. If the hypothesis is true, then this is at least true for bad designs—precisely the ones where their being ranked too high could have the most negative impact.
- Furthermore, if it is true, then the study would also show a way to run the studies that avoids the problem; namely, don't just show one design—rather, show three (or more). That way users are not placed in a position of being negative, since they can balance any negative comments about one design with positive comments about another.
- The real significance of paper interfaces and other low-cost techniques in usability testing would be shown to be that they make such parallel testing of alternative solutions possible. However, this is not the common practice, nor what is taught in the textbooks. The hypothesis being true could help change this.
- The hypothesis being validated would highlight the need to maintain a design mentality; that is, pursue various alternatives, rather than zero in on a particular solution, throughout the process, including usability testing.

Now I've told you what the hypothesis is, and the significance if it is true. The next thing to do is figure out if it is true. The good news is that the answer is determinable. The way of doing so is to follow the example of Wicklund *et al.* and run an appropriate study. Such is the progress of science.

We ran such a study in the summer of 2005 (Tohidi, Buxton, Baecker & Sellen 2006a). In fact, it was in the course of that study that Maryam built the paper interface shown in Figure 145. But what you have seen so far is only one of three versions that she built. The other two are seen in Figure 151. As you can see, each is in a distinct design language, or style, but all three are rendered at about the same level of resolution, and have the same functionality.

Figure 151: Two Alternative Programmable Climate Control Interfaces

These interfaces are functionally equivalent to the one shown in Figure 145, and the associated figures
that follow. However, each utilizes a different design language, or style. The one in Figure 145 uses a
dial-based interface. The one on top in this figure uses a tabular form-based approach, whereas the one
at the bottom uses a time-line based approach.

Briefly, subjects performed the same set of task on all three interfaces, one after the other. The order in which the three were presented was changed in order to counter any bias that might influence the results. After having used all three, they filled out a questionnaire, for them to rate each.

In parallel with this, we also ran another set of comparable subjects who saw only one version of the interface. They ran the same set of tasks as those who saw all three, and were asked to fill out the same post-task questionnaire and evaluate the interface.

The quick summary of the results, at least as they pertain to our current discussion, is that the hypothesis was shown to be true. If you compare the ratings given to the lowest rated interface of the three, as judged by subjects who had seen all three, that rating was significantly lower than the rating given to that same interface by subjects who saw only it.

I have already given you four reasons why we should care about this result. But I would like to close this discussion with two overriding, related conclusions:

This study helps emphasize why we should not commit to a design too soon, and why there is value in continually exploring various options to any question.

The most important value in using these quick and inexpensive techniques is not that they save money in making a prototype (although that is not to be sneezed at). It is that they make it affordable to make and compare alternative design solutions to problems throughout the design process.

This second point is not just true for the paper-based techniques but virtually all the techniques covered in this section of the book.

Finally, there are inherent potential problems in some of what we have been speaking about, such as presenting users with so many options that they are confused or overwhelmed. What I am trying to do is build up a basic literacy around sketches and prototypes. Understanding what makes each distinct is an essential step toward making the right choices in an environment otherwise dominated by the prototype approach.

In other words, our purpose is to provide a counter-balance to statements like:

For most applications, an evolutionary, whole-system, continuous prototype is a desirable choice for the user interaction developer. (Hix and Hartson 1993, p. 256)