Bill Buxton brings design leadership and creativity to Microsoft. Through his thought-provoking personal examples he is inspiring others to better understand the role of design in their own companies."
Bill Gates—Chairman, Microsoft Corp.

Sketching User Experiences

getting the design right and the right design

Bill Buxton

MK MORGAN KAUFMANN

Publisher: Diane Cerra
Publishing Services Manager: George Morrison
Senior Project Manager: Brandy Lilly
Editorial Assistant: Asma Palmeiro
Copyeditor: Adrienne Rebello
Indexer: Distributech Scientific Indexing
Interior printer: Transcontinental Interglobe
Cover printer: Transcontinental Interglobe

Book Design: Henry Hong-Yiu Cheung
Information Graphics: Henry Hong-Yiu Cheung
Typography Consultant: Archetype
Illustrations: Adam Wood

Morgan Kaufmann Publishers is an imprint of Elsevier.
500 Sansome Street, Suite 400, San Francisco, CA 94111

This book is printed on acid-free paper.

Library of Congress Cataloging-in-Publication Data
Buxton, William.
Sketching user experience : getting the design right and the right design / Bill Buxton.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-12-374037-3 (pbk. : alk. paper)
ISBN-10: 0-12-374037-1 (pbk. : alk. paper) 1. Design, Industrial. I. Title.
TS171.B89 2007
658.5'752–dc22
2006036416

ISBN 13: 978-0-12-374037-3
ISBN10: 0-12-374037-1

For information on all Morgan Kaufmann publications, visit our Web site at www.mkp.com or www.books.elsevier.com

Printed in Canada
08 09 10 11 5 4 3 2

Figure 78: Workshopping Ideas
One of the best ways to draw out the best from people, designers and users alike.
Photo: Brooks Stevens Design

From Thinking On to Acting On

...we are in danger of surrendering to a mathematically extrapolated future which at best can be nothing more than an extension of what existed before. Thus we are in danger of losing one of the most important concepts of mankind, that the future is what we make it.
—Edmund Bacon

Now we change gears.

What we are going to do in this part of the book is look at the work—and more particularly, the working methods—of very good designers, from established professionals to talented students. This approach serves five important functions:

- *To illuminate what I perceive as best practices.*
- *To help those who work with the design team (including managers and the executive team) to understand these practices and their output.*
- *To foster a shared literacy among the design team of some of the relevant "classic" examples from our diverse traditions.*
- *To show exemplary student work side by side with that of those who pioneered the field in order to show that what I am advocating is attainable.*
- *To give a sense of some of the basic competencies that I would expect in an interaction/experience design team, and hence in the educational programs that train them.*

When I speak of "best practices," I am referring to a repertoire of techniques and methods with which I would expect any Experience Design team to have a reasonable degree of fluency. This is not a "How to design a great product" manual, or a treatise on "How to be creative," but it does stake out part of that turf, namely a subset of design primarily relating to ideation and sketching. There is a good chance that someone who reads this section will be familiar with some of what I discuss, but I suspect that there will be few for whom there is not something new. And, even with familiar material, I hope that I am able to bring a sufficiently fresh perspective to contribute new insights.

As for the second point, before product managers or executives dismiss the material in this section as being irrelevant to them, they might want to recall Alan Kay's quote that I mentioned earlier:

> It takes almost as much creativity to understand a good idea, as to have it in the first place.

One of the best steps toward fostering a common culture of creativity among a diverse team is to become as fluent as possible in each other's languages. I have tried to make this book as accessible to the businessperson as the designer because I think that the designer's efforts will be for naught if the executive and product manager don't understand the how and what of the designer's potential contribution to the organization. Just think back to the case of Jonathan Ive at Apple. Do you want to squander the potential of your design team, as was largely the case until Steve Jobs came back to Apple, or do you want to improve your ability to exploit it the way that Steve did? Simply stated, the sooner you understand a great idea, the more lead time you have to do your part in executing it. That is why you need to read this section. Not to become a designer, but so that together with your design team (which you are paying for anyhow!), *and with the rest of your organization*, you can make design a more effective differentiator in your company.

As to the third point, I confess to being captivated by history— of my profession and of almost everything I am interested in. To me, history is both interesting and part of basic literacy. I think that it is important to the effective practice of our craft. The problem is, the experience design team of today involves people from many different traditions, each with its own history. I would hope that those from each tradition would know their own history, but I would never assume that they know each other's. For example, industrial designers will likely know about Christopher Dresser (Whiteway 1991), Norman Bel Geddes (Bel Geddes 1932), Henry Dreyfus (1955), or Raymond Loewy (Tretiack 1999), and why they are important. But more often than not, these names will draw a blank when given to a user interface designer who has a computer science or psychology background. By the same token, names such as Doug Engelbart (Bardini 2000), Ivan Sutherland (Sutherland 1963), and J.C.R. Licklider (Waldrop 2001), which should be familiar to the user interface designer, are most likely unknown to those from the tradition of industrial design.

Yet, the histories of each of our various disciplines, including marketing, have the potential to lead to more informed design. Knowing each other's histories lays the foundation for shared references and the common ground that that creates. So, whenever appropriate, I have chosen to mix key historical examples from various traditions into what follows. Although it is not a history lesson per se, hopefully it will make some contribution toward building a shared literacy and tradition among the emerging culture of experience design.

Fourth, while familiarity with some of the classic examples from our history is important, it can also be intimidating. By relying on such examples, am I setting the bar too high? Is this standard attainable by a student, or is this too much to ask from someone that you are thinking of hiring? I think not. I have consciously also incorporated examples from the work of students from around the world to convince you of this point. For me, meeting these students and being exposed to their work was one of the most encouraging and enjoyable parts of researching this book.

Finally, a new approach to design implies a new approach to design education. Let's say that what I talk about makes sense and that by some miracle executives all over the world say, "Yes! Let's incorporate something like this in our company." Who are they going to hire? Where are the people going to come from? What kind of skills and experience should one be looking for? This section provides the basis for a partial answer. But I need to add, yet again, a cautionary note: This is not a comprehensive manual on product design. I am only trying to fill a gap in the literature, not cover the whole space. There are other books on topics such as Participatory Design, User Centred

Design, Usability, Industrial Design, Ethnography, Marketing, and so on. We do not have to start from scratch. Second, no individual will or can have equal competence in all the requisite skills. So the second thing to keep in mind is that we need coverage of the larger skill set distributed among a heterogeneous team, not the individual. But, and this is the important "but," for that team to function well, the players must have at least basic literacy in each other's specialties, if not a high level of competence.

Is this section going to be technical? On the one hand, yes, we are going to dive into the design funnel and talk about what goes on inside. On the other hand, it is not going to be any harder to follow than what we have already discussed. And I certainly hope that it is as interesting and relevant. It is definitely not going to take the form of some academic analysis of formal design theory or methodology. Why bother? As Chris Jones says in his book, *Design Methods*:

> There seem to be as many kinds of design process as there are writers about it. [There is] little support to the idea that designing is the same under all circumstances, and ... the methods proposed by design theorists are just as diverse as are their descriptions of the design process. (Jones 1992; p. 4)

In many ways, we wouldn't be in our current situation if formal design theories and methodologies worked as advertised, with their many boxes and arrows that map out the process. Gedenryd (1998) makes this argument pretty well. Speaking about architecture, Snodgrass and Coyne (2006) say:

> Contemporary architecture theory now largely ignore the vast literature on systems theory and design methods ... (p.24)

And in his book, How Designer's Think, Bryan Lawson remarks:

> Well, unfortunately none of the writers ... offer any evidence that designers actually follow their maps, so we need to be cautious.
>
> These maps, then, tend to be both theoretical and prescriptive. They seem to have been derived more by thinking about design than by experimentally observing it, and characteristically they are logical and systematic. There is a danger with this approach, since writers on design methodology do not necessarily always make the best designers. It seems reasonable to suppose that our best designers are more likely to spend their time designing than writing about methodology. If this is true then it would be much more interesting to know how very good designers actually work than to know what a design methodologist thinks they should do! (Lawson 1997; p. 39)

Whenever possible I have video clips that compliment what I say with words and pictures. These can be accessed from the companion website to this book: www.mkp.com/sketching.

I have structured this section in a kind of musical "E-A-B-C-D" form. Perhaps this is my earlier life as a composer coming out. I am going to start with a few rich examples that foreshadow where we are going, then pull back to a simpler world. From there I will build back up toward where

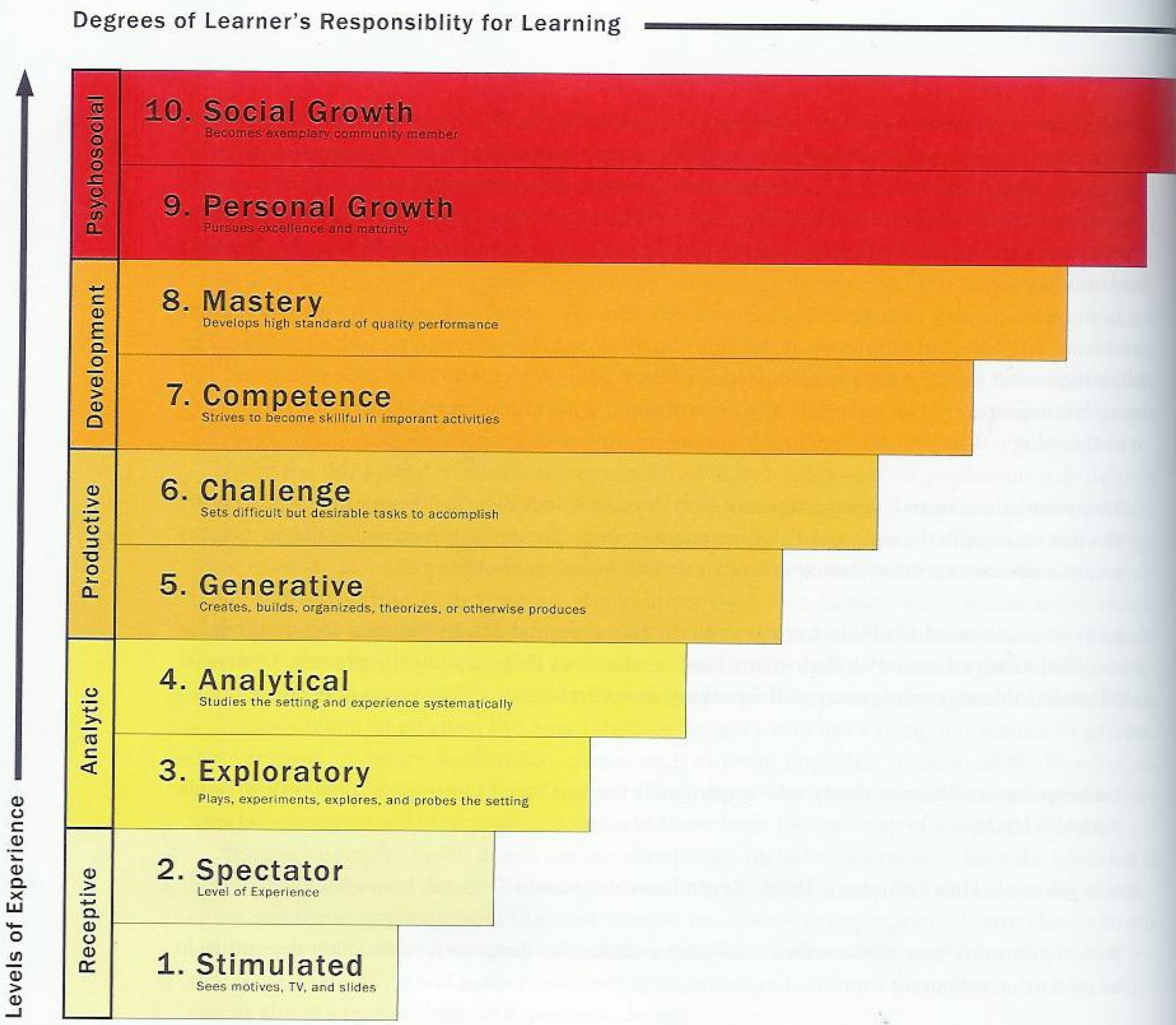


Figure 79: Scale of Experience in Learning

Ten levels of increasing experience in learning are shown. As the level increases, the learner takes on additional responsibility.

we started, laying more of a foundation in the process. And just as a warning, somewhere in the middle, I am going to insert an interlude where I can add some meta-comments and examples.

But when I talk about richness or space, what is the scale on which my A, B, C, and so on lie? I am going to draw on a tangentially related field, experiential learning (see Kolb 1984, for example). In this literature, Gibbons & Hopkins (1980) developed a *Scale of Experience*, illustrated in Figure 79. With it, they attempt to establish a kind of taxonomy of levels of experience. Although a legitimate target for debate, it can serve our purpose.

At the lower levels are things where one is at the receptive end of experience. The notion is that although you can experience seeing a train or a bear in a movie, there is a higher level of experience seeing it live. Likewise, there is a deeper level still if you get to play with the train or (hopefully Teddy) bear, rather than just see it. The argument made is that as one goes up the scale, one moves through different modes, from Receptive through Analytic and eventually through to what they call Psychosocial mode.

If we push too hard on this its relevance to our work diminishes. After all, the scale was developed for a different purpose—education rather than design. There are really only three things that I want to draw out of it.

First, when I say that I am going to organize this section on an E-A-B-C-D structure, I am going to start with a few examples from the high end of a scale analogous to that of Gibbons & Hopkins. I will then drop back to examples and techniques that are at the lower, receptive, level of the scale, and work my way back up.

Second, Gibbons and Hopkins argue that higher levels of experiential learning imply a higher level of responsibility on the part of the learner for what they learn (the auto-didact). This is represented by the horizontal axis in Figure 79. Likewise, from the design perspective, our renderings (be they sketches or prototypes) afford richer and richer experience as we go up the scale. However, reaping the potential benefit of the design knowledge, or learning, that can be extracted from these renderings also depends on assuming the responsibility for using them appropriately.

Third, going a step further from the previous point, keep in mind that the level or type of experience that one can get out of renderings at the lower levels should not necessarily be considered impoverished. Seeing something live is not necessarily better than seeing it in a movie—it is just different. There are different types and levels of experience. Knowing how to use them appropriately in design is where the artistry and technique come in.

Finally, before proceeding, I want to point out that I did notice the "Those who can, design, and those who can't, write about design" aspect of the earlier quote by Lawson. The irony of including it, much less Lawson's writing it in the first place, is not lost on me. I have tried to keep its message in mind in what I write. Second, I think that there are times that design goes through transitions due to new demands that are made on it. In such times, thought and writing about design can provide a useful role in helping us get through those transitions with minimal problems. I view us as being in the midst of just such a transition, hence my sticking my neck out and taking up my proverbial pen.

The Wonderful Wizard of Oz

I can't leave the future behind.
—Suze Woolf

Perhaps the most important book in terms of informing our endeavor is *The Wonderful Wizard of Oz* (Baum 1900). To some this might seem a strange statement. But think about our purpose: it is to experience interactive systems, before they are real, even before we have arrived at their final design, much less their implementation. Keeping that in mind, think again about Baum's book, or the film version of it, and consider this:

> **Up to the point where Toto tipped over the screen and revealed the Wizard to be a fraud, all of Dorothy's reactions were valid psychologically, anthropologically, and sociologically. To her the Wizard was real, and therefore so were all her experiences.**

In the Introduction to this book I said something that read like a paradox:

> **The only way to engineer the future tomorrow is to have lived in it yesterday.**

What Baum's book teaches us is that if we do an effective job of following the example of the Wizard, we too can conjure up systems that will let users have real and valid experiences, before the system exists in any normal sense of the word. In all of this, the four most important things to glean and carry forward are:

- It is fidelity of the experience, not the fidelity of the prototype, sketch, or technology that is important from the perspective of ideation and early design.
- We can use anything that we want to conjure up such experiences.
- The earlier that we do so the more valuable it generally is.
- It is much easier, cheaper, faster, and more reliable to find a little old man, a microphone, and some loud speakers than it is to find a real wizard. So it is with most systems. Fake it before you build it.

I'm not sure who first used the term "Wizard of Oz" in the context of interaction design. I first heard it from one of my early influences, John Gould, of IBM. But regardless of who coined the term, the meaning is pretty well understood internationally (unlike the story): the *Wizard of Oz Technique* involves making a working system, where the person using it is unaware that some or all of the system's functions are actually being performed by a human operator, hidden somewhere "behind the screen."

The objective is not to make the actual system, but to mock up something that users can actually experience, thereby enabling us to explore design concepts in action and as experienced far earlier in the process than would otherwise be possible. Such a system should be cheap, quick to realize, disposable, not the real thing, and only have sufficient fidelity to serve its intended purpose. That is, it should have all the attributes that characterize a sketch.

Inherent in all this is the following rule:

Generally the last thing that you should do when beginning to design an interactive system is write code.

Now I know that a very large proportion of interactive products are software driven, and that the status quo is that software engineers play the primary role in their development. Having gone through this myself, I also know the amount of time and effort that these same software engineers have expended in acquiring their skills. I even know that they or their parents probably spent a fortune paying for that education. But I also know the following rule:

If the only tool you have is a hammer, you tend to see every problem as a nail. (Abraham Maslow)

Having gone through all the effort and expense to develop these skills, the natural inclination is to damn well use them. And the better you are at it, the more likely it is that this will be the case. After all, that excellence probably came at the expense of not developing other skills (since there are only so many hours in the day and nothing comes for nothing). So the point here is that all the biases in the status quo stack up against the Wizard, and in favour of programming and engineering. Not just from the individual perspective, but also from that of management, who have made a huge investment in the best programming talent available. ("What? I hire all these great computer scientists and engineers and now you are telling me not to use them?")

It is precisely because of the magnitude of this investment that we must make a conscious effort to manage things along a different path. These resources are too expensive to squander on a product that is almost certain to fail—and fail it will most likely do, if one leaps into engineering prematurely.

I hope that it is clear that I am not trying to downplay the importance of programming and engineering skills. I am simply saying that the front-end of the interactive design process is generally not the appropriate place to apply them. Furthermore, the need to balance the biases of the status quo is yet another reason that design, and not engineering, should own this front part of the process. As I have said previously, engineering is culturally as unsuited for the task of managing design as designers are for running engineering. With product engineering in charge of the front end of the process, the temptation to fall back into old habits will be just too great to resist.

So if we are not going to start by programming, and we do want to follow the Wizard, then some examples may help shed some light on how to do so.

Airline Ticket Kiosk

Historically, the first instance of using the *Wizard of Oz Technique* that I am familiar with was in the field-testing of a self-service airline ticket kiosk concept (Erdmann & Neal 1971). In the past few years, these kiosks have appeared in most airports, but in 1971 that was not the case. In fact, computerized public kiosks were not yet familiar to the general public, much less ubiquitous—this was the same year that the first automated teller machine was deployed. Compared with today, even credit cards were not that common. Consequently, there were few, if any, precedent technologies on which design decisions could be based. So they built a prototype, and after testing it in the lab, field-tested it at Chicago's O'Hare Airport.

Since it was being used by real customers, using real money to buy real tickets on actual flights, it had to work from both the user-interaction perspective, as well as that of the back-end ticketing and financial system. But the system wasn't finished. That was the whole point: they wanted to test it before deciding to proceed with the product, much less commit to a final design.

So they went straight down the Yellow Brick Road and faked it, like any good wizard. To the user, everything looked real. But behind the curtain, there was a human being, acting as The Great Oz, who keyed in the pertinent information using a conventional terminal, and then issued the ticket. Although expensive and inefficient from the short-term perspective of actually delivering the service to customers, the potential value in terms of informing the design, usability, and acceptability of a potential future system was immeasurable.

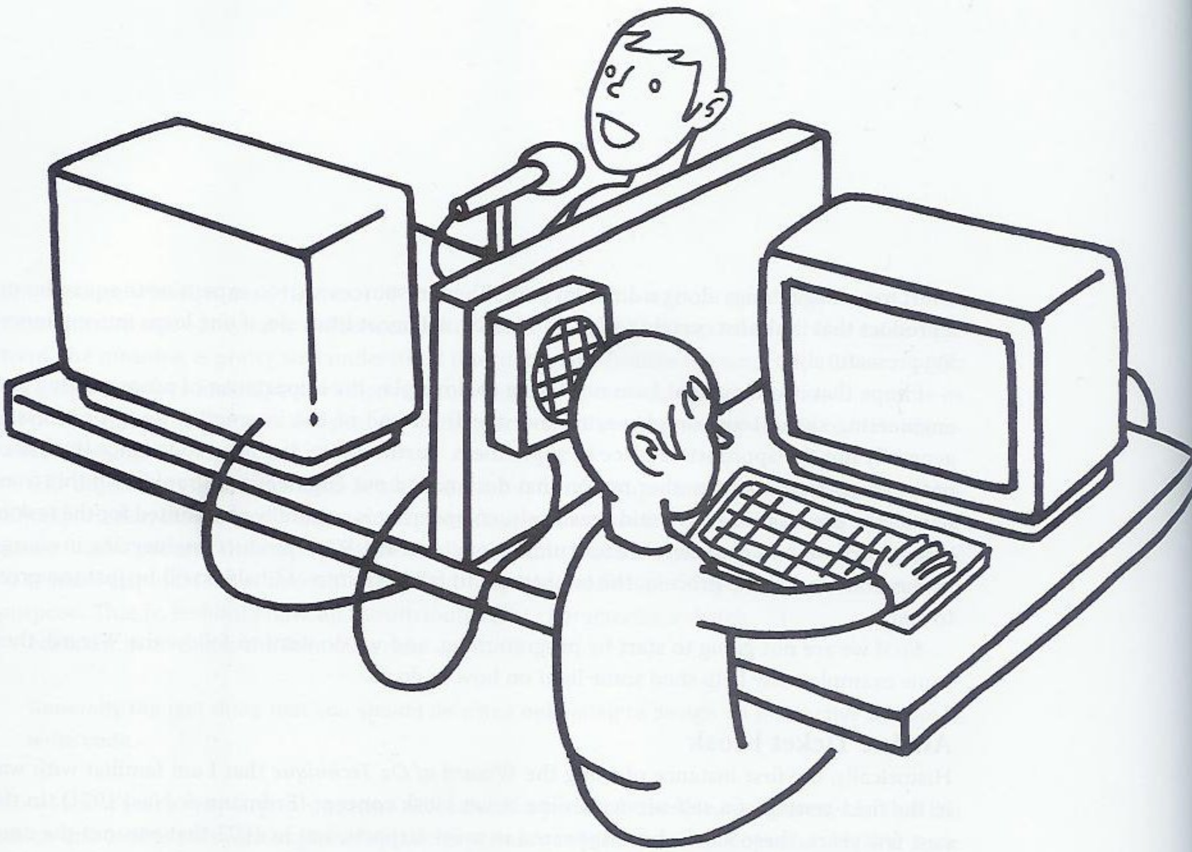


Figure 81: The Wizard's Listening Typewriter

A perfectly functional listening typewriter is implemented simply by having a fast typist, hidden behind the screen, who would enter the text captured from the microphone.

The Listening Typewriter

Practically everyone involved in user interface design has had at least one person come up to them and explain how, "If we could just talk to computers like we talk to people, and if they could understand what we are saying, then the user interface problem would be solved."

However, just because people have been saying such things for a long time does not make their assertions true. On the other hand, it doesn't make them false either.

To shed some light on the matter, John Gould and his colleagues at IBM decided to perform a study to investigate the potential benefits of a listening typewriter (Gould, Conti, & Hovanvecz 1983).

At the time, the late 1970s and early 1980s, IBM had a lot at stake. They had a major position in office information systems, and they wanted that position to consolidate and grow. If there were potential benefits to be derived from a listening typewriter, then they wanted to be the ones to reap them. The problem confronting them was how to understand what such benefits might be, especially when the underlying technologies did not yet exist. They could spend a fortune on R&D and wait years for it to work sufficiently well to support a realistic test. But what if the expected benefit wasn't there?

How could they make an informed funding decision that was more of a calculated investment than a gamble? Here is where Gould and his colleagues stepped in, and engaged the Wizard in order to test the system before it existed.

The way that they did this is illustrated in Figure 81. Their approach was as ingenious as it was simple and effective. Playing the role of the Wizard was a speed typist, who really did hide behind the curtain. Whatever the user dictated into the microphone, this "typing Wizard" entered into the computer such that it appeared on the screen of the user. Not surprisingly, the system worked fantastically well. Words were properly spelt, and paragraphing and punctuation matched the speaker's intention. In fact, that system probably worked better 20 years ago than any real system does today. Using the *Wizard of Oz Technique*, they were able to leap more than 20 years ahead in the technology curve, and collect real user experience data! And, they were able to do so in a matter of weeks, without writing any substantive code. This paper is a classic, and should be studied by students of interaction design the same way that art students study the classics of the renaissance.

Figure 82: An Example of a Display That Changes According to Location

A camera's viewfinder is an example of a display whose content changes depending on what you point it at. Even pointing at the dog, what you see changes as you move the camera left-right, up-down, or in-out.

Chameleon: From Wizardry to Smoke-and-Mirrors

Only a god can tell successes from failures without making a mistake.
—Anton Pavlovich Chekhov

One of the people who learned from Gould and his colleagues was one of my PhD students, George Fitzmaurice. Like others, he realized that one could create the illusion of a working system using techniques other than having a human Wizard perform some of the functions of the system. In some cases, such as the example that we are going to look at, the magic, or illusion, comes by means of the clever use of technologies and techniques on the part of the designer. Rather than a human Wizard, "smoke-and-mirrors" technologies are applied to realize an interactive sketch of a concept in a form that users can actually experience.

In the early 1990s, we were investigating

> ... how palmtop computers designed with a high-fidelity monitor can become spatially aware of their location and orientation and serve as bridges or portholes between computer-synthesized information spaces and physical objects. (Fitzmaurice 1993).

I know that that was a mouthful, so here is a simple version of the question he was asking: What if the contents on the screen of a handheld device could be determined by what the device was near, what it was pointed at, or how it was moved up-down, left-right, or in-out?

Imagine looking through the viewfinder of a camera, such as that illustrated in Figure 82. It has all these properties. If you point it at a dog, you see the dog. If you move it closer, the image of the dog gets larger on the screen, but you see less of it. If you pan the camera left, you see the dog's tail in the viewfinder. If you pan it to the right it shows his head, and so on.

With the camera there is no magic. The screen just reflects what is in front of the lens. But what if the device in your hand was not a camera, but a PDA, for example? What if instead of panning across a dog, as in our example, you were panning over a large spreadsheet or some other document? What if you were doing so simply by moving the PDA left-to-right as if you were operating a camera? Likewise, what if when you came up to an object in a museum and the PDA could "know" what was in front of it, and display additional information on the artifact depending on where you pointed it?

George called this type of position and motion sensing display, *Chameleon*. He wanted to gain some sense of what it would be like to use such a system. How would it feel? Would people be comfortable using it? What would the experience be compared to that of using a desktop computer?

These are simple questions to ask (at least once you have come up with the idea). In order to answer them, "all" that he needed was a handheld computer that sensed where it was in space, and the computational and graphics power of the most powerful workstations of the day.

Figure 83: A Small LCD TV Masquerading as a Position-Sensitive PDA

All we could afford was this little LCD TV display. It worked, so it was all that was needed. Anything more would have been superfluous.
Source: Fitzmaurice (1993)

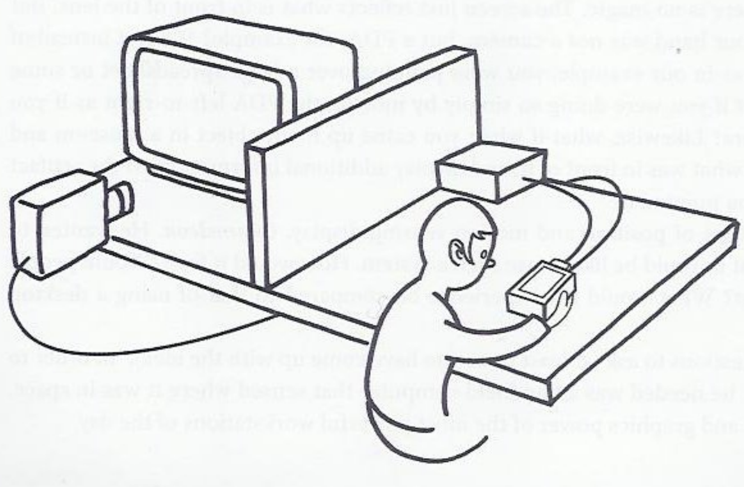


Figure 84: Faking a Position-Sensitive PDA

The scenario is much like that in Figure 72, except this time the Wizard is replaced by a video camera and other miscellaneous technologies that we could borrow or scrounge.
Source: Fitzmaurice (1993)

But there was this minor problem. He was doing all this in 1992, a year before the Apple *Newton Message Pad* was announced and four years before the release of the original *PalmPilot*. Nevertheless, he was able to implement a system that let one personally experience this type of interaction. Here's how.

To begin, he bought the small $100 Casio LCD TV shown in Figure 83. This served as a surrogate for the "handheld computer" with which future users might interact.

So far, so good. But this was just a TV and had no computer power. However, what it did have was a video input. So, George hooked it up to a (borrowed) video camera and pointed the camera at the screen of a (borrowed) state-of-the-art SGI graphics workstation. That caused a video image of the computer display to appear on the Casio TV. *Voila!* He now had the makings of palmtop computer with the computational and graphics power of an SGI workstation.

He then had to figure out how to sense the position and motion of the LCD display as it was moved around by the user. To do this, he borrowed a device called a Bird, made by Ascension Technologies. This is what is known as a motion-capture device, originally designed for computer animation. It consisted of a small cube that George unobtrusively attached to the back of the Casio TV, and it then connected to the SGI, where it provided the spatial information required to drive the interactions seen on the handheld display. The overall configuration is shown in Figure 84.

Admittedly the device was tethered, and therefore did not have the mobility that we envisioned in the future. Nor did it have the graphics resolution that we anticipated would become available in the future. But by using it with some 3D software that he borrowed from another project, it did let George get a good first approximation of what it would be like to interact with such a device, and more to the point, it let him observe the experience of others doing so.

It also enabled him to explore different types of applications and transactions. Figure 85, for example, illustrates panning across a display. Unlike a desktop computer, where you use the scroll bars and scroll arrows to move the document on a stationary display, here you move the display over the surface of a stationary virtual document in a manner analogous to our camcorder example.

Likewise, Figure 86 illustrates how the behaviour of the *Chameleon* can be driven by its position over an object in the physical world, rather than some virtual document. In this case, the object is a map. But when coupled with the *Chameleon*, the paper map becomes a guide for browsing for more detail about its surface. So, for example, if you move the display over Winnipeg, Manitoba (where George is holding it), it might give you details about the city's important role in the fur trade, or why it is the mosquito capital of Canada.

Figure
The us
held d
using
Sourc

Figure
The ph
map c
handh
the de
Sourc

Figure 85: Panning Across the Display
The user navigates across the information space by moving the hand-held device much the same way that you would pan across a scene using a camcorder.
Source: Fitzmaurice (1993)

Figure 86: Chameleon in Context to the Physical Environment
The photo illustrates how something in the physical world, in this case a map of Canada, can work with Chameleon. For example, by bringing the handheld device into the vicinity of southern Manitoba, as illustrated, the device would provide information about Winnipeg.
Source: Fitzmaurice (1993)

In short, by employing smoke-and-mirrors techniques, George was able to initiate research into a new class of interaction, in a way that enabled him and his users to gain direct personal experience, at least 10 years before such interfaces were commercially viable.

So, is George's *Chameleon* example a sketch? I want to spend a bit of time on this question since it provokes some issues that run through this part of the book.

First, I'm not sure. Given the tools and resources available at the time that it was done, it probably lies somewhere just to the left of centre on the sketch-to-prototype continuum.

Second, perhaps the real value in drawing a marked distinction between sketching and prototyping lies not in the end points, but in recognizing that there *is* a continuum between them. An awareness of it, its properties, and its implications, may help guide us in how and when we use different tools and techniques.

Third, as we shall see later in our discussion of "paper prototypes," how a technique is used is the ultimate determinant of whether one is sketching or prototyping. At the time that it was done, the *Chameleon* project would most accurately be described as "rapid prototyping." Yet later, I give a more recent example where I explored the concept in a context that I would call sketching.

Finally, regardless of label, I think that our repertoire of techniques needs to incorporate the facility and mindset reflected in George's example.

Having said that, I also want to address one thing missing in his work. As agile as his approach was, he could not take the interactive result (rendering) out into the field to test. Its size and complexity meant that in order to use it, people had to come to the lab.

There are trade-offs and consequences of the decisions that we make, and we have to be aware of them. But if we are aware of them, we can incorporate techniques into our plan that help compensate for any shortcomings. For example, notice how George dealt with this particular issue. Through the use of staged photographs taken in situ, such as that shown in Figure 86, he was able to connect the dots between what one could experience live in the lab, and various visions of its application "in the wild." (See his original paper for further photographic and diagrammatic examples.) This is a nice example of how various media (including the accompanying video) complement each other in order to tell the larger story.

The lesson here is that it is rare that only one form of rendering will suffice, which is one of the reasons that the design team must have command over a range of techniques. I am not going to give any magic rules or guidelines when or where to use which technique. Far more effective is for you to get practical fluency with the techniques that we discuss. The experience gained in the process will provide far more effective insights about their relative strengths and weaknesses

Figure 87: A 2D Chameleon Sketch

By sticking a piece of paper under a Tablet-PC to reduce friction and attaching a mouse to the side, one can quickly make a working sketch of what it is like to navigate around a virtual map by sliding the tablet around on the surface of a desktop, like a moving window.

than anything that I could write. Words are not sufficient to provide such understanding. It must be based on experience.

Finally, I want to revisit George's rendering of *Chameleon* with today's eyes, and in particular, the tools and technologies that are now at our disposal. Recently I was involved in a project to explore new concepts for lightweight slate (tablet) type computers. Since we were talking about ways to navigate over maps and other documents that are too large to entirely fit on the screen, I thought of George's example.

What I was able to do, right in the meeting, is render a working interactive sketch of what I was talking about: a desktop version of *Chameleon* that worked in 2D. First I placed a piece of paper under my Tablet-PC so that when I slid it over the table surface, there was minimal friction. I then loaded a map program on the computer and scaled it to the full screen size. Finally, I stuck a miniature wireless mouse to the side of the PC. I was then able to navigate over the surface of the map by moving the tablet on the desktop, thereby realizing a version of George's *Chameleon* idea. This is illustrated in Figure 87 and the associated video.

Yes, I had to hold the left mouse button down, and yes, I had to have the mouse-tracking symbol over the map surface for it to work. And so as not to hide anything, I also had to adjust things like the mouse control:display ratio and orientation to get the feel that I wanted. In a matter of minutes everyone in the room understood the basics of the concept, as well as some of the issues underlying its implementation. Furthermore, despite some in the room already having heard or read about similar ideas, everyone left with a renewed appreciation for the difference between cognitive versus experiential understanding.

Not only was this a sketch (meeting all the criteria that we identified), it was the experience that was sketched. Furthermore, it was done on the spot in the wild, so to speak.

One quick final comment to emphasize the notion that design has as much to do with attitude as technique: What is significant, but not obvious, is that I shot the video clip that illustrates this sketch right in the meeting. I always carry around a compact digital camera with a really big memory card with which I can collect reference images of things that interest me, as well as shoot videos, such as this one. Hence, I can share material with others on the team, even if they are not there to experience it first-hand. This "hunting and gathering" of reference material is a habit that borders on mania with every good designer that I have met. We shall return to this theme again.

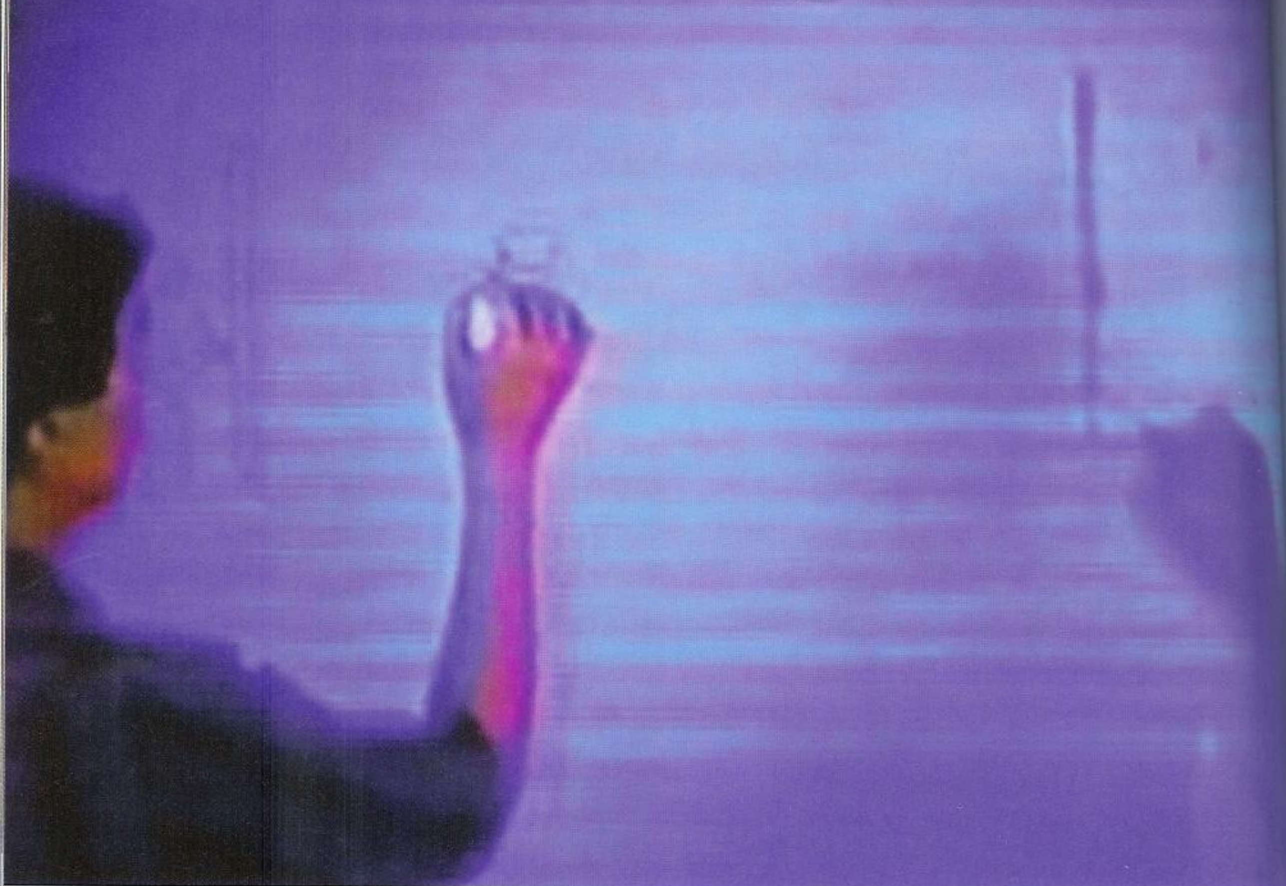


Figure 88(a): Video Whiteboard

This is a shared drawing surface on which the marks created by both the local and remote person are combined. Furthermore, the remote person has a presence by virtue of a life-size shadow that appears to be just on the other side of the surface, regardless of how many thousand miles away they might be.

Source: Tang & Minneman (1991)

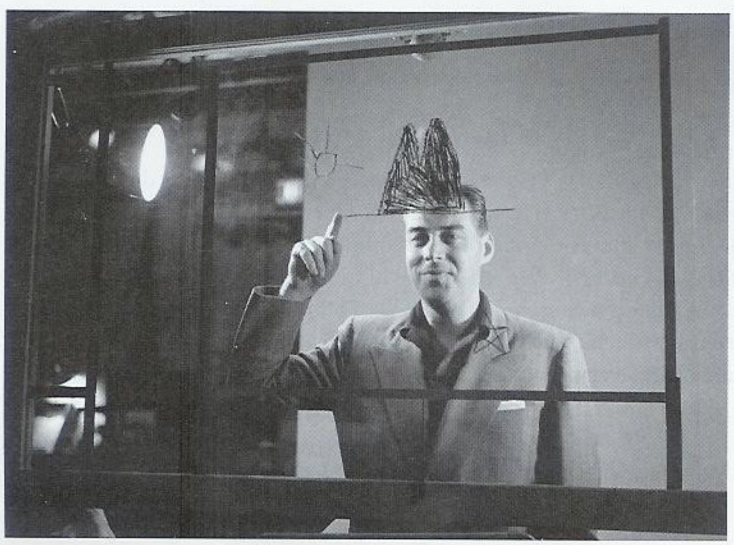


Figure 88(b) Winky Dink and Remote Drawing

Refer back to Figure 66 of the TV show Winky Dink and You. This is a shot of what was going on in the TV studio. It illustrates how the host and child could both draw together. Other than the minor fact (!) that the host couldn't see the child or what was being drawn at the other end, it is a wonderful example of how popular media can contain the seeds of future ideas, and be rich terrain for relevant reference material

Photo: CBS.

Le Bricolage: Cobbling Things Together

Are you experienced?
—Jimi Hendrix

I now want to complement the previous examples with one that exudes the elegance, simplicity, and effectiveness of the best of sketching. What I love about it is that it is highly interactive, provides a very high fidelity experience, is a surrogate for a futuristic technology, yet involves no computer at all in its implementation. It was made using materials found in most modern offices or universities.

The Video Whiteboard

The project is *The Video Whiteboard* and it was done by John Tang and Scott Minneman at Xerox PARC (Tang & Minneman 1991). They were interested in collaborative design, where the collaborators were separated by geographic distance. The scenario that they envisioned was a large electronic whiteboard at each location, which provided each participant a surface on which to draw, as well as to see what the other person was drawing.

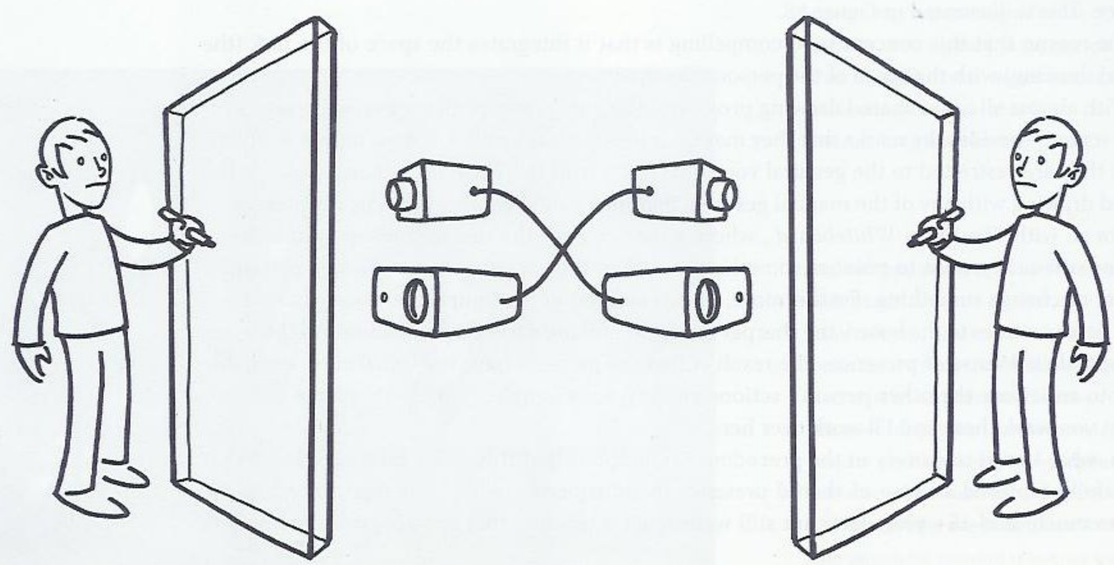
To this point, the idea is interesting, but not that special. Many people have imagined and even built surfaces for shared remote drawing. What was special with Scott and John's vision was the idea that you could see the shadow of your collaborator on the surface, as well as what they drew, and that it would be life size, appearing as if the person was just on the other side of the surface. This is illustrated in Figure 88.

The reason that this concept is so compelling is that it integrates the space of the task (the shared drawing) with the space of the person (the shadows).

With almost all other shared drawing programs, the only presence that a person has on a remote screen—besides the marks that they make—is the tracking symbol of their mouse or stylus. Thus, they are restricted to the gestural vocabulary of a fruit fly! Thus, they cannot refer to the shared drawing with any of the manual gestures that they would employ if physically present.

Not so with *The Video Whiteboard* , where a user can see the remote person approaching, such as when they want to point at something, or when they are about to work in a particular region, or change something. Furthermore, as can be seen in the figure, the closer that the remote person comes to the board, the sharper the focus and the darker their shadow. All this gives an unparalleled sense of presence. The result is that the partners have the visual cues to enable them to anticipate the other person's actions and say, for example, "No, don't change that!" or "Okay, you work there and I'll work over here."

So, what I tried to convey in the preceding paragraph is that this was a pretty cool idea that beautifully captured a sense of shared presence in collaborative work. The fact that I like the idea so much (and 15+ years later am still waiting for a product that even begins to deliver its



potential) only heightens my reason for including this example. For, as interesting as the idea is, how they sketched it out is just as good or even better.

Now if you had this idea yourself, perhaps the fastest way to "build" a sketch of it that you could experience might be to do what I have illustrated in Figure 89. That is, just get two people to stand on either side of a window or glass door and have them draw on it using conventional whiteboard markers.

This is a good start (and something that I encourage you to do, rather than just read about). But, it only partially captures the concept. What one gets is a "full fidelity" image of the remote person, not the shadows that John and Scott wanted.

The next step would be to do the same thing, but this time using glass that is fogged. That way you can get a very fast sense of the trade-offs of using shadows versus full-fidelity representations of the remote person. (For an account of a parallel project that uses the full fidelity approach, see Ishii & Kobayashi 1992.)

After doing these quick proof-of-concept sketches, you can explore the case where the participants are separated by more than just a sheet of glass. Here is how Scott and John did it, and in so doing, won my undying respect.

Their solution, illustrated in Figure 90, involved each person having their own rear projection screen on which (like in the previous implementations) they drew using conventional whiteboard markers. Pointed at the back of each screen was a video camera that captured what was drawn on the screen, as well as the shadow of the person doing the drawing (by virtue of the ambient back-lighting). The captured video was then projected onto the back of the screen of the other user.

Augmented by a bidirectional audio link, the two screens (and therefore the two people) can be as far away from each other as you like, and still appear to be just on the other side of the glass. And, in the implementation, there is even automatic control over permissions: each user can erase only what they themselves drew.

The reason that I like this example so much is its simplicity, and the way that one can quickly implement and iterate through the experience sketches that I have described. It is a great idea, as relevant today as when it was first conceived. It is *very* relevant to collaborative computing, and yet there is not a bit of computer or digital technology involved in its implementation. In fact, I would argue that doing so would most likely distract one from the basic idea, rather than help.

For me, the *Video Whiteboard* is one of the classics that should be studied and replicated by any serious interaction designer.

Figure 89: First Iteration: Nothing but Glass

The first iteration in a video whiteboard is simply having people drawing on either side of a sheet of glass or Plexiglas. All that is needed is something like a window (such as a glass door) and whiteboard markers.

Figure 90: Video Whiteboard Schematic

What is drawn by the person on the left, as well as their shadow, is captured by the video camera on the left. This is then projected onto the back of the other screen, where it is visible to the person on the right. Likewise, the person on the right and what they have drawn is captured by the right camera, and projected on the screen on the left.

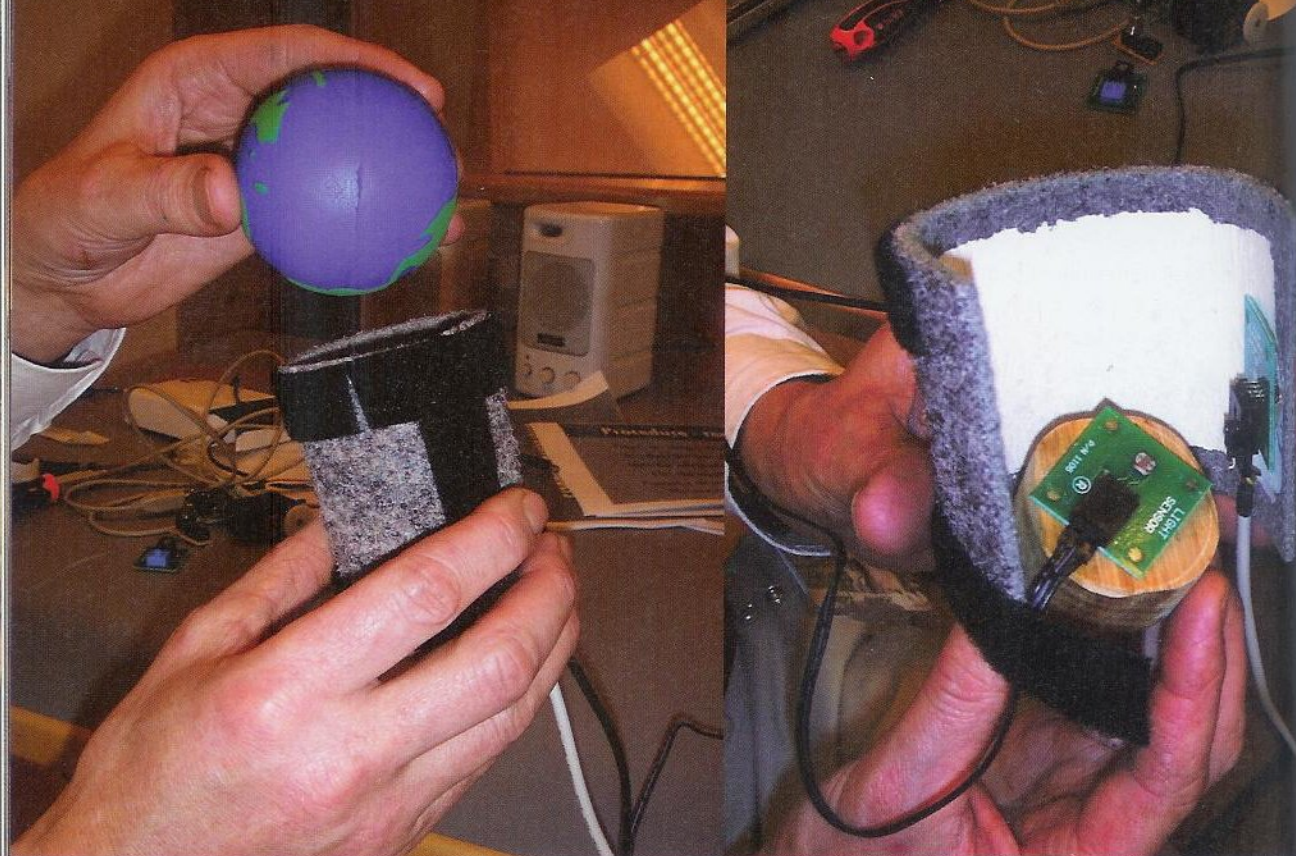


Figure 91: A Working Concept Sketch of a Gesture-Controlled MP3 Player

Gesture-Controlled MP3 Player

I stated that, "the tools and technologies that are available to us today are opening up new ways of approaching things." One such example is the emergence of tool-kits that let us rapidly assemble hardware and software components to build working devices, much in the way that we assembled LEGO bricks to make things when we were kids. Of course, one always could build things with miscellaneous hardware (electronic and otherwise) and software. The difference today is that doing so is within the reach of normal people, and does not require the specialized skills of an electrical engineer and computer scientist.

Here is an example that illustrates the kind of thing that is now within the reach of the design team.

In April of 2006 Abi Sellen and I invited Caroline Hummels and Aadjan van der Helm of the ID-Studiolab in Delft to conduct a workshop on sketching using physical prototypes for the group that we are part of at Microsoft Research Cambridge. Aadjan and Caroline arrived with multiple bins of materials and tools, as well as a broad assortment of *phidgets* (www.phidgets.com).

The task was to build a working MP3 player. We split into four teams of three or four, and had to design and implement a player that was appropriate for a particular persona—all in one day—including learning to hook up the hardware, program it, and iterate through the design. Oh, and one rule: the programmers were not allowed to program, and hardware people were not allowed to do hardware. The idea was to demonstrate that in short order, using the right tools, making such things was within the capability of social scientists and industrial designers, for example.

Figure 91 shows the player made by my team, which included the social scientists Richard Harper, Andrew Fogg, and Martin Hicks. As is illustrated in the first image, one opened up the world of music, so to speak, by lifting the ball off the top of the device, which was made of fabric in a cylindrical form. The volume was controlled by how much the top was uncovered. To change tracks, one punched in a new track by thrusting the device forward. To change genres, one shook the device up and down. Not something that is going to make anyone a million dollars, but it worked.

The second image shows the guts of the device, a photosensor on the bottom, and a 2D accelerometer on the side. The whole thing was programmed using *MAX/XSP* and Jitter (www.cycling74.com), and an MP3 module supplied by Caroline and Aadjan.

The *MAX* program that controlled our player is shown in Figure 92. It is divided into three parts, each represented by a horizontal region in a distinct shade of blue. The middle section, labelled "Product Functionality," is the heart of the MP3 player—the part that actually plays music. This module was provided to each team by Caroline and Aadjan. It does the work, but has no external user interface.

The bottom section, labelled "Product Feedback," was also provided by them. It provides the kind of visual information that you would expect to see on the display of a CD player or your car radio.

Finally, the top layer, labelled "Sensor Mapping," is the part that we "wrote" to implement our particular gesture-controlled player. The top-left box in this section, labelled *r ai2*, is an icon that represents the source of the numerical data that *MAX* receives from the light-sensor that we want to control volume. By drawing lines out of the bottom of this box and connecting them to the top of another, we can make the current value "flow" from one box to another. In a perfect world, to control volume using this sensor, we would just have to draw a line from the bottom of this box to the top of the box labelled *Volume* in the *Product Functionality* section. But we don't

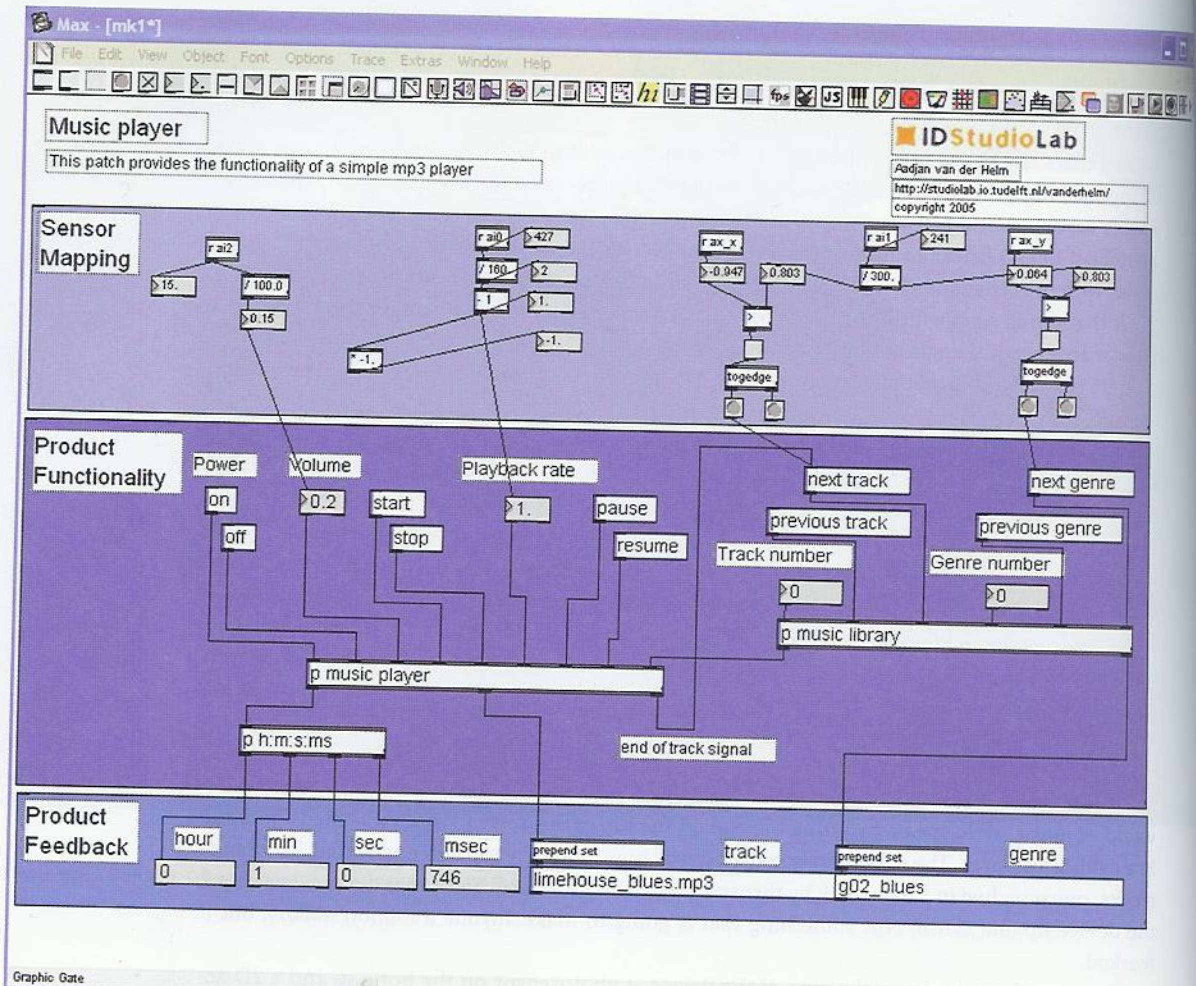


Figure 92: Program to Control the Gesture-Controlled MP3 Player

live in a perfect world and being true to life, our exercise was not that simple. The range of values output by the sensor were far too large for the MP3's volume control. For example, if the volume control went only from 1 to 10 but the range of our sensor produced numbers from 1 to 100, we would be able to use only 10% of the full dark-to-light continuum.

We knew that the range of the volume control was 0 to 10. So our first step was to see what the minimum and maximum values coming out of the light sensor were. To determine this, we added a numerical display box that would show us the instantaneous value coming out of the sensor. This is the box to the lower left of the r ai2 box. It shows us that at the time that this figure was made the output was 15.

Experimentation told us that the range of values from the light sensor was 0 to 1000. Consequently, we passed the sensor's output through an arithmetic box, which divided the incoming value by 100, thereby enabling us to use the full dynamic range of the sensor, and still remain within the proper range of the volume control.

That this operation worked can be seen in the numerical display connected below the division, which shows that the value has been changed to 15. This is the value that is then piped to the input of the *Volume* control.

We then used similar techniques to map the outputs of the other Phidget controllers to the Playback Rate, Next Track, and Next Genre controls of our MP3 player.

Stepping back and looking at the *Wizard of Oz*, *Smoke-and-Mirrors*, and *Bricolage* examples that we have discussed, I have to concede that their roots are mainly in research rather than product development. The fact is, this is simply where the techniques originated (at least in terms of the publicly available literature).

Being sensitive to the not infrequent gap between research and product cultures in organizations, I want to emphasize that these roots in no way disqualify the use or relevance of such techniques in product design. In fact, many innovative designers will likely yawn at some of my examples as old hat, having used them for years.

There are two general take-away lessons here. The first is the way of thinking about and exploring the design ideas that these examples represent. The second is hinted at by my modern return to *Chameleon*. It gives us our first clue that the tools and technologies that are available to us today are opening up new ways of approaching things. Not only is the barrier-of-entry to using these techniques dropping really fast, but those same changes mean that we are frequently able to take these things out of the lab and into the wild, whether for user testing or participatory design. (For more on this approach to physical interfaces, see Greenberg & Fitchett 2001, Holmquist, Mazé & Ljungblad 2003, Holmquist, Gellersen, Kortuem, Schmidt, Strohbach, Antifakos et al. 2004, Greenberg 2005 and Villar, Lindsay & Gellersen 2005.)

There may be as many approaches to sketching interaction as there are products to design. In this volume we will scratch only the surface. But we have to start somewhere.

These have been my opening "E" examples. Now we are going to drop down to the "A" level on the experience scale, and work our way back up.