3

From the Desk to the Palm

Interviews with John Ellenby, Jeff Hawkins, Bert Keely,
Rob Haitani, and Dennis Boyle

In the 1990s there will be millions of personal computers. They will be the size of notebooks today, have high-resolution flat-screen reflexive displays, weigh less than ten pounds, have ten to twenty times the computing and storage capacity of an Alto. Let's call them Dynabooks.

Alan Kay, 1971[1]

Desktop workstation from Metaphor Computers, designed by Mike Nuttall of Matrix Product Design and Jim Yurchenco of David Kelley Design

Photo
Rick English

Designing the laptop and the palmtop was about shrinking the computer so that you could take it with you, first as a luggable suitcase, then in your briefcase, and eventually in your pocket. The transition from desktop machines to laptops was about designing the physical interface to be small enough to carry easily without changing the interactions on the display significantly because of the smaller size. And the same interface and applications had to work on both.

Alan Kay is well known for his summarizing the Xerox PARC credo as: "The best way to predict the future is to invent it!" His conceptual contributions helped in the formation of the Alto and the Dynabooks in parallel, so we look first at how his ideas about portability emerged, and how his group at PARC developed the first luggable computer, the NoteTaker. The next dramatic shrinking of the machine was the leap from luggable to laptop. The author's personal account of the story of designing the GRiD Compass, the first laptop computer, is included in the introduction, as it triggered his quest for interaction design. The rest of the story of how the Compass came into being is told in an interview with John Ellenby, the founder of GRiD Systems.

treo
handspring
powered by msn
News
Stock Quotes
Sports
Yellow Page
Directions
Weather
Directory of Mobile S
Google S
Copyright © 2001 Hand

Jeff Hawkins joined GRiD as an entrepreneurial young engineer and, after gaining some experience working for John Ellenby, was given responsibility for developing GRiDpad, the first tablet computer to reach the market. He describes how he went on to create the palmtop, leading the team that created the Palm operating system (OS) and developing the series of PalmPilots that were so successful. The Palm OS was developed in 1995 and was an immediate success. It is dramatically different from the operating systems that evolved for personal computers: there is no mouse or desktop. Rob Haitani was a key member of Jeff's team right from the beginning and was in charge of the interaction design for the Palm OS and its applications; he discusses the reasons behind the design of the operating system in his interview and recounts the details of the process.

Jeff Hawkins went on to found Handspring, extending the range of products using the Palm OS. He had worked with Dennis Boyle of IDEO to develop the Palm V, a design that was attractive enough to dramatically increase the popularity of electronic organizers, and continued to work with Dennis for a series of products at Handspring. Dennis tells how the Palm V was designed and how he and his team developed a camera to plug into the Handspring Springboard slot. The chapter concludes with the story of the development of the Handspring Treo, which combined the PDA organizer with an email communicator and a cell phone.

■ Handspring Treo

Photo
Courtesy of
Handspring

Alan Kay

Alan Kay argues that user interface design started when the people who were designing computers noticed that end users had functioning minds. He describes his own epiphany:

> For me it was the FLEX machine, an early desktop personal computer of the late sixties designed by Ed Cheadle and myself. Based on much previous work by others, it had a tablet as a pointing device, a high-resolution display for text and animated graphics, and multiple windows, and it directly executed a high-level object-oriented end-user simulation language. And of course it had a "user interface," but one that repelled end users instead of drawing them closer to the hearth.[2]

In 1968 he felt that he was "hit on the head" by several amazing innovations. At the University of Illinois there was the first little piece of glass with glowing text characters, giving a glimpse of the future potential for flat screen displays. He then read McLuhan's *Understanding Media* (1964),[3] and the concept that "the medium is the message" made him believe that it is in the nature of people to be reshaped by tools, that the invention of the printing press really did make us a scientific society, and hence living in the age of computers will reshape us again:

■ Alan Kay

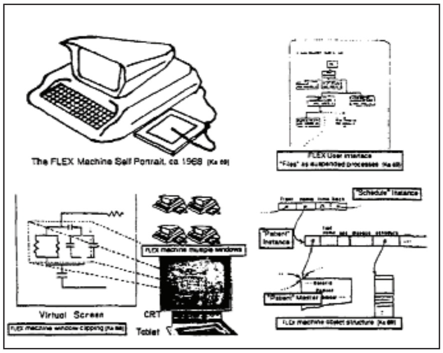
Photo
Courtesy of
The National
Academies

> The computer is a medium! I had always thought of the computer as a tool, perhaps a vehicle[4]—a much weaker conception. What McLuhan was saying is that if the personal computer is truly a new medium, then the very use of it will actually change the thought patterns of an entire civilization. He had certainly been right about the electronic stained-glass window that is television—a remedievalizing tribal influence at best.[5]

Shortly after reading McLuhan, Kay visited Seymour Papert at MIT's Artificial Intelligence Laboratory. Papert had spent five years working with Jean Piaget, a psychologist who studied the ways in which children learn, and had found that they could explore computer-simulated worlds as they explore the real world. Alan saw children writing computer programs that generated poetry, translated English into Pig Latin, and created arithmetic environments. Before that he had thought that people would need to be able to program a computer before they could feel that they owned it, but here was a demonstration, even with children, that as long as you can "read" a medium, you can make use of and enjoy material and tools that are made by other people, and if you can "write" in the medium, you can create material and tools for other people:

> If the computer is only a vehicle, perhaps you can wait until high school to give "driver's ed" on it—but if it's a medium, then it must be extended all the way into the world of the child. How to do it? Of course it has to be done on the intimate notebook-sized Dynabook! But how would anyone "read" the Dynabook, let alone "write" on it?

This was a moment of inspiration that changed Alan Kay's life. He became fascinated by the idea of developing computers for children and started on a quest in that direction that still engrosses him. It is disappointing that this decision has kept him an abstract contributor in the world of the laptop and palmtop, as the earliest products to be developed for manufacture and sale were inevitably high priced,[6] and therefore aimed at the people who could afford to pay the most, such as executives, spies, and soldiers, rather than children. The portable machines that Alan developed and inspired were laboratory prototypes and much less influential than Smalltalk and his other software contributions. In spite of this, he is recognized as the father of the laptop.


The FLEX Machine Self Portrait, ca. 1968 [Ka 69]
FLEX User interface
"Files" as suspended processes [Ka 69]
"Schedule" Instance
"Patient" Instance
"Patient" Master
FLEX machine multiple windows
Virtual Screen
CRT
Tablet
FLEX machine window clipping [Ka 69]
FLEX machine object structure [Ka 69]

- Dynabook concept
- FLEX Machine

Learning

The idea that learning involves more than one state of mind has emerged from Alan's thinking as a basic principle of interaction design, applying to adults as well as children. After the visit to MIT, he started to study child psychology and learning processes, and he discovered the work of Jerome Bruner.[7] Bruner described three distinct stages in the learning process, each of which has a different state of mind: "Enactive, iconic, and symbolic." This was illustrated by an experiment with two water glasses, one short and fat, and the other tall and thin. When children are shown the contents of the short glass being poured into the tall one, they will say that there is more water in the tall one, even though they saw the pouring. If you then hide both glasses, the children change their minds, reasoning that the water had nowhere else to go. This could be done several times; each time the children would repeat the assertion that there was more water in the tall thin glass whenever they could see it but deny it when it was not visible. The experiment illustrates the different mental states that underlie our learning process.

> Bruner convinced me that learning takes place best environmentally, and roughly in stage order—it is best to learn something kinesthetically, then iconically, and finally the intuitive knowledge will be in place that will allow the more powerful but less vivid symbolic processes to work at their strongest. . . .
>
> Now, if we agree with the evidence that the human cognitive facilities are made up of a doing mentality, an image mentality, and a symbolic mentality, then any user interface that we construct should at least cater to the mechanisms that seem to be there. But how? One approach is to realize that no single mentality offers a complete answer to the entire range of thinking and problem solving. User interface design should integrate them at least as well as Bruner did in his spiral curriculum of ideas.[8]

What I hear, I forget.

What I say, I remember.

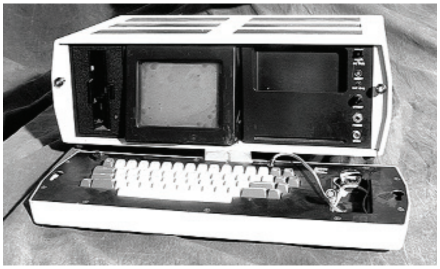
What I do, I understand.

Lao Tse

Designers are used to the idea of learning by doing. They know that creative work is more likely to succeed when they experiment before they try to understand, moving from doing things to creating an image of a design intention and then iterating this process until they recognize the value of the resulting design. Only later, or perhaps never, do they understand the symbolic relationships of the solution that they have created. Understanding these three stages could help us design interactions that allow a similar learning process.

Luggables

Alan Kay inspired the researchers at PARC by promoting a long-term vision that he called "Dynabook." The Dynabook was to be a computer so portable and so connected that a pupil would carry it between home and school daily, instead of carrying printed textbooks and paper notebooks. Around 1970 Alan drew pictures of what, today, we would think of as small laptops, large PDAs and tablet computers. In 1976, having developed Smalltalk, he shifted his attention toward a new project called NoteTaker, another notebook-sized computer. The central idea was to take a percentage of the Alto's functionality and put it in a compact portable machine. The design was to include a custom-built display screen that was touch-sensitive—to eliminate the mouse—stereo audio speakers with a built-in microphone, a rechargeable battery, and an Ethernet port. However, the technology at the time was highly limited. The actual NoteTaker was a plump attaché case that looked like the first generation of the "luggable" computers that were created six years later. The lid flipped down with the screen and disk drive set in the body, which faced the user when the box was laid flat on a table. The lid held the keyboard, which was connected to the computer inside the body by a flexible cable. Larry Tesler was part of the team at that time. Here's his description of the process.

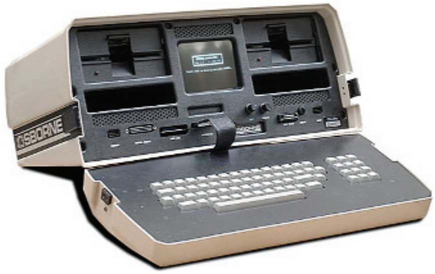


NoteTaker

> For the first time in my life, I got involved in a hardware project. With a little help, I designed the CPU board. I also figured out a way to squeeze an Ethernet controller on a single, small circuit board, something the hardware engineers thought was impossible at that time. The NoteTaker was an amazing machine. It contained three 8086 processors. (We got early samples of the 8086 from Intel in 1978.)
>
> It was heavy—35 pounds plus 10 pounds if you inserted the batteries—but it was the most powerful portable computer of its time, if not the only one in existence. After we got it running, and Smalltalk was ported to it, we vowed to convince the Xerox management to fund development and mass production. We scheduled trips to demo the system to various Xerox executives. Unlike the Alto, which had to be shipped as freight, we were able to carry the NoteTaker with us on the plane. Once, when nobody was around, we turned on this top-secret machine in an airport lounge and ran a Smalltalk program. This may have been the first time a portable computer was used in an airport by a business traveler.

Ten prototype machines were built, but that was not enough to convince the Xerox management to produce them. The director of service got excited about the idea of every technician who serviced copiers replacing a big stack of blueprints and manuals with a portable computer. His vision was that they could connect the NoteTaker to a modem and download the repair manuals for the copier that was being serviced or repaired. He was willing to back the production of the design but never got top management approval.

The most famous luggable computer was the Osborne, which was released in 1981. The tiny five-inch diagonal screen displayed white characters on a dark green background and was symmetrically flanked by two 5¼″ floppy drives. It weighed almost twenty-five pounds without the battery pack and when closed looked like a sewing machine case. The symmetry gave it a strangely facial expression when open, with the floppy drives like armored eye slits, and the cable that connected the keyboard like the tongue of a cheeky little boy. Priced at $1,795, the Osborne 1 was wildly successful at first, selling more than eleven



Osborne ■
Compaq 2 ■
Dynalogic Hyperion ■

thousand units in the first eight months. The boom was short lived, as the IBM PC was starting to dominate the market, and Osborne did not bring out an IBM-compatible machine in time to prevent bankruptcy in 1983.

The demand for "luggables" was filled by the Compaq Portable, running MS-DOS on an Intel 8088 processor, and with a nine-inch diagonal screen that could display twenty-five lines of eighty characters. Priced at $3,590, this design served Compaq well, and by 1986 they were on the Fortune 500 list, breaking a speed record in arriving there. The Dynalogic Hyperion was another MS-DOS machine, released three months earlier than the Compaq. The keyboard could be slid underneath the main unit and locked into place for carrying.[9] It lost the battle with Compaq because it was not fully PC-compatible and had some reliability problems with the disk drives. It vanished after two years.

These "luggable" machines were battling for market share in the same timeframe as the GRiD Compass computer, which was released in 1982 at a price of $8,150. The story of how that led the author toward the design of interactions is told in the introduction. Next, John Ellenby, the founder of GRiD Systems, gives his account of how it happened.[10]

Photo Author

John Ellenby holding the briefcase that determined the size of the first laptop

When John Ellenby arrived at Xerox PARC in 1974, he was surprised by the thing called a mouse; he was used to a solid-feeling trackball to manipulate the cursor on the big graphic displays at Ferranti in the UK, where he had spent the previous few years working on a multiprocessor system for communications and process control, while simultaneously holding a faculty position in the Computer Science Department at the University of Edinburgh. He was tenured as an assistant lecturer at the London School of Economics when he was still in his twenties, using computers as a research tool. Xerox invited him to PARC, hoping that his combination of academic background and experience in industry would help in the transition from research findings to real products. He was impressed by the research and the people and enthusiastically signed up to build the first twenty-five Altos. He went on to reengineer the Alto and to develop Dover, the first preproduction version of a laser printer. In 1977 he was given responsibility for "Futures Day,"[11] a demo to show off the achievements of the PARC researchers to the entire top management of Xerox. John teamed up with Tim Mott to make it spectacular. He hired talent from Hollywood to make sure the presentations were professional and arranged for the majority of the prototype equipment at PARC to be shipped to Boca Raton in Florida for the event. It was a huge success, but Xerox remained focused on the reproduction of documents, causing an exodus from PARC. It was then that John decided to found his own company to create a new kind of computer.

GRiD
Compass Computer
Quarterly Projections
7000
6000
5000
4000
3000
2000
1000
10
20
30
40
Group A
Group B
Group C
GRiD

John Ellenby

Developing the First Laptop

GRiD Compass computer preproduction prototype, 1981

Photo
Don Fogg

While John Ellenby was at PARC he got to know Alan Kay and appreciated the promise of the Dynabook concepts. At the same time, he was worried by Alan Kay's goal of developing machines for children. John was pragmatic enough to know that the initial price point would be too high for normal kids for anything as innovative as Alan was describing. The starting point had to be the customer with the most money and the most demanding need. Here is his description of the conversation that inspired the laptop:

> I was meeting with a guy in the government. He was actually in the executive office of the White House. He said, "I really like the Alto, John. I really like the Dover laser printer. I really like this gateway to the ARPAnet. I really like that. It's really good. I get emails this way, and documents this way, and the documents fly backwards and forwards, and this is really terrific and I really love it!"
>
> And I said, "Well, do you have anything that you don't like?"
>
> "Well, I don't use it!"
>
> "What do you mean? You said you like it."

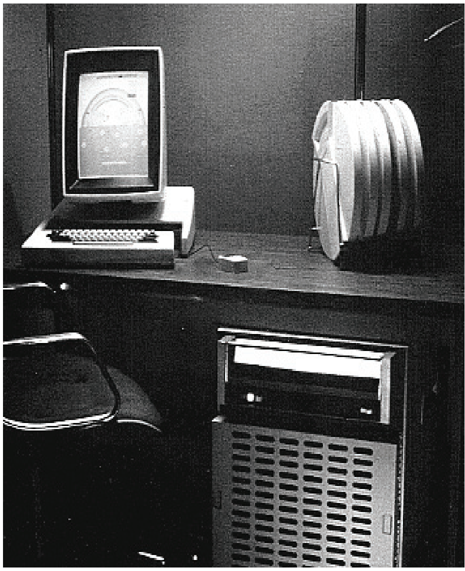
> "I learnt enough about it very quickly, it's very simple to use. I've got my secretary using it, and she's a bright person, and she's the one that uses it."
>
> "Is that how you think it's going to go forward?"
>
> "No. Let me tell you what I want. I want all this capability of your beautiful Alto, and I want it in half your briefcase."
>
> And he went and grabbed my briefcase, which I actually still have. It was a classic one for a Xerox executive—leather, with a solid wooden liner. I said, "This briefcase? That's hard!"
>
> He said, "I want it to go in half that briefcase, because I carry papers around."
>
> "Do you think you are typical of people?"
>
> "Yeah, let me tell you. My job as a senior member of this establishment is to go to where the problems are. It's not my job to hang around in the office. I have staff here. I call them on the phone, I get to them by fax, I send them telexes: do this, do that, and they do it. My job is to go to where the problems are, so I want a computer, a real computer (he actually referred to it as an RFC—you can guess what the F is for), I want a real something-or-other computer to carry around with me that has the communications, has all the capabilities of the Alto, but is downsized for me to carry."
>
> Boy, that was a real opening for me, because at Ferranti I'd been looking at the technology of flat panels and the technology of microprocessors, and nonvolatile storage, and I was projecting out when all of those would come together into something that you could carry. Because I had lots of people that wanted them for fire control and process control—somebody who wants to go out and look at the process and also look at the thing that's controlling it. So to be able to link, and ideally by wireless, with something tablet like.

It was the briefcase that did it. It had to fit in the briefcase; not just inside, but in half of it! That became the design brief for developing the GRiD Compass computer. It was a seemingly impossible challenge that appealed to John's imagination and set him going with boundless energy and determination. He had seen the components used in computers steadily shrinking, but if you piled them on a table they would still amount to a volume that was probably double that of the briefcase, not half. He would have to attack each individual component, either completely reconfiguring the item for a new shrunken configuration or thinking of a lateral solution that would allow him to use a completely different item.

First he had to start a company to do this, so he left Xerox in January 1980 and started structuring the dream team to make it possible. He gathered key Silicon Valley figures for his board, successfully raised money, and put together an amazingly talented team of founders for GRiD Systems. The company was structured in the best Silicon Valley tradition of attracting people of the highest caliber by offering an exciting challenge and founders' stock. They would then be motivated to dedicate their lives to the success of the company, working long hours and doing everything themselves to begin with, with the expectation of becoming the highest officers in the company when the product proved its value. There were key players in charge of R&D, manufacturing, marketing, and experts in electronic hardware, software, and physical engineering and design. John transformed himself as he got the company going. In three months he changed from a free-living and easygoing person to someone who was dedicating every second of his life to a single purpose.

One of the challenges was to find a way to hold the data and avoid the bulky disk drives that were being used for machines like the Alto. The Alto certainly sported a beautiful high-resolution display and mouse for interacting with the software, but the rest of the machine was huge and clumsy, too large to fit on top of a desk, let alone in a portable unit. John called it the "Gzunda."

> I called mine Gzunda, which no one understood. In Britain Gzunda is the potty you put under the bed. People said, "Why Gzunda?"
>
> I said, "Well, it goes unda the desk."
>
> It occupied the whole of the desk, so you had to sort of cramp your way in. You had to dedicate a large part of your office to this box. Much of it was taken up by a big Diablo disk drive that held almost nothing at all, but in those days, that was all we could get. The processor was big and chunky, and the screen and everything else needed a lot of work.



Alto

The answer was to use bubble memory, a nonvolatile storage component from Intel. This could then be combined with modem access to remote data, to be provided from servers at "GRiD Central," and each individual customer could have access to a larger file capacity than the 256K of bubble memory that was built into the laptop. The modem was also an essential part of the communications solution, but modems were big then too.

> We managed to get a modem built. We went and met with Racal Vadic. We said we wanted a 1,900-baud modem, and they said, "Oh yeah, we make those. We'll go get one."
>
> They brought this thing out that's twice the size of a shoebox. We said, "Yes, we want a 1,900-baud modem, and we want it to use only 55 watts, and we want it to be this size," showing them something the size of two packets of cigarettes.
>
> "No way! Absolutely no way."
>
> "Well, we've spec'd it out, we have to have a 1,900-baud modem, so either you're going to do the work for us, or someone else is gonna do it."
>
> "No, no, we can't do that!"
>
> Our hardware guy was sitting across the table from his counterpart in Racal Vadic, who gave him this wink, and we thought, "This guy had obviously figured out that, yes, with a monolithic, they know how to do it."
>
> So we engineered a sort of around-the-corner discussion with this guy, and sure enough, they could do it. It was a wonderful example of the Valley working.

The electronic components were starting to look somewhere close to feasible. They settled on the Intel 8086 processor. The details of the size tradeoffs would drive the physical design, with the most obvious questions being the size and weight. The first question for the component arrangement was which half of the briefcase to fill; a blockish volume could fill the left or right half, and a large flat arrangement would fit in the top or bottom half. The large flat alternative was advantageous for the keyboard and display—and an elegant proportion.

The first keyboard was too wide and too thick, so they developed a layout that still had the QWERTY arrangement but made do with fewer additional keys. They then found a vendor who was be willing to develop a thinner construction that still had good human factors of key travel and snap action for tactile feedback. The first display that they found was too small, using a limited number of characters on five lines, so that the amount of information that you could see was disappointing. They eventually found a prototype of a bright electroluminescent display of tiny golden yellow squares, developed by Sharp in their labs in Osaka, with individually addressable pixels rather than an array of characters. John describes the negotiations with Sharp:

> The small screen that we had for the launch was an amazing screen; it was done for us by Sharp. We found it through a wonderful Japanese guy called Glen Fukuda, who came in and said, "I don't know anything about electronics, but this is exciting, and I understand you need a screen, and where would you go? Obviously Japan. And where would you go in Japan? Well, let's go talk to these companies."
>
> He did a bit of research. It was wonderful to work with this guy. He convinced Sharp Corporation, who were working on electroluminescent panels in the lab, to come and meet with us. I think it was Mr. Okana, who was the head of all research at Sharp Corporation, who came out first. He came to my house in Palo Alto and had to walk under this enormous oak tree. It had been raining, and the rain was still dripping from the oak tree. Afterwards his translator told Glen Fukuda to give the message to Mr. Ellenby that Mr. Okana, "Visited, knew the future, and raindrops fell, even though it was not raining."
>
> I still get goose bumps when I think of that. This guy was head of Sharp's research. He had decided some time back to take the money that they would have used for a huge show of all the Japanese electronics companies and create this research laboratory for the EL panel and other advanced components. There was this wonderful moment when Sharp committed to build the electroluminescent panel for us. It was gorgeous, a wonderful, beautiful orange; I'd never seen anything like it.

The six-inch diagonal seems tiny now, but it was amazing to find such an excellent flat display at the time, which supported bitmap graphics as well as characters. Sharp built a factory

John Ellenby with the GRiD Compass computer at launch ■

IBM PC ■

especially to manufacture it for GRiD. All this was very new at the time, as there was not yet even an Osborne, an IBM PC, or an Apple Mac, let alone a pervasive Internet.

The power supply needed to deliver a whopping eighty watts, and was a huge brick at the beginning, so they worked on breaking it down and integrating it with the main printed circuit board to reduce the volume.

The first calculation for the weight was over eleven pounds, enough to stretch your arm if you carried it around for long in a briefcase that already had papers in it. The founders of the company were equipped with sets of weights, and asked to adjust the amount that they carried around in their briefcases so that it was starting to hurt, but without being completely intolerable. They came to a consensus that there was a dramatic cross over point at eight pounds, so that became the weight specification.

John wanted a repair service that would benefit from the overnight shipping offered by Federal Express, so that customers could send back a faulty machine and have it returned to them with a two-day turnaround. This set a standard for a drop test that was more demanding than the accidents that were likely if the machine fell off a table or was dropped by someone carrying it. They rented an impact recorder to find out how much rough handling the shipping would entail and sent it by Federal Express to Washington and back. When it returned, they unfurled the chart paper of the recorder in a long snake across the floor of the office. They saw the 20g impact (twenty times the force of gravity) when the recorder was collected from the office and tossed into the truck. They saw about the same impact when it was loaded on to the plane at the airport, but the big shock was at three in the morning when it was put through the automated sorting system at the central office of Federal Express in St. Louis. There it saw 60g, the equivalent of dropping it from three feet onto a concrete floor. This then became the specification for the robustness of the enclosure, a demanding mechanical engineering challenge. They chose to design the enclosure in die-cast magnesium, for the combination of strength and lightness, succeeding in meeting the specification and creating an amazingly tough machine. It was sent up in the space shuttle and used on Air Force One as a result.

The GRiD Compass computer was launched to the public at the Office Automation Conference in the spring of 1982 and was highly acclaimed. The price was double that of the luggable machines, so the market was limited to cases where high value was attached to small size, prestige, performance, and durability. The plan was to sell the products to large companies, so that the individuals could gain access to company-wide information on the road, connecting in to the "GRiD Central" server with the built-in modem. The most common users were expected to be sales and business executives. This audience did prove to be the mainstay for the company, but the business climate for computers was dramatically altered when IBM unveiled their personal computer in August 1981, and by the time the Compass was out, the world was moving fast toward MS-DOS and PC-compatibles. GRiD OS had a set of central common code that was very compact and supported a spreadsheet, a text processor, a terminal emulator, a message/email solution, and a draw program—all with a consistent user interface. They also had a universal file system structure so that a call from a program would go directly to the data, wherever it was. It could be in RAM, ROM, bubble memory or local disk, external local disk, out over wire at "GRiD Central," or over the GRiD server within a client company. These were strong advantages, but not enough to resist the surge of demand for IBM PC compatibility, so John decided that the next version of the Compass would have to run MS-DOS, and it would be needed fast.

> In order to do an MS-DOS machine, we had to do an IBM-compatible machine that ran MS-DOS. I remember going up to visit with Bill Gates; Charles Simonyi was up there at the time. Charles and I had worked at Xerox together. We showed them what we were doing, and Bill says, "This is great. Of course we'll do a special version of MS-DOS for you."
>
> I could have reached across the table and given this guy a kiss.

Pictures of GRiD Compass in space

GRiD GRiD
GRiD GRiD

The speed with which the GRiD team created the new version to run both GRiD OS and MS-DOS probably saved the company, which avoided the fate that had overtaken Osborne. The extraordinary toughness of the design also gave the unit an appeal for intelligence agents and soldiers, and soon GRiD was doing a thriving business in Washington. John talks about soldiers playing games:

> We were selling a lot of GRiD computers to the armed forces. We had this wonderful game called Flak Attack. These tanks shot at each other and at airplanes—a great game. You selected the device and fired the gun by hitting the spacebar. We had a lot of equipment with the ninth infantry division in Fort Lewis, Washington. They bought a whole lot of GRiDs because they were developing mobility concepts—how can you move fast, run around in ATVs—a pretty exciting outfit. They took them out to the Yakima firing range and used them out there, which is sort of high deser—very cold at night, very hot during the day, very dusty. They were charging around with their GRiDs, and several GRiDs came back. I thought that this was a problem. It turned out that all of them had broken space bars. We thought, "Have these guys been playing Flak Attack?"
>
> They were getting carried away. Whack! Whack! They'd actually been smashing the spacebar like proper infantry men should. You know, FIRE, FIRE, FIRE, FIRE! So that was a relief.
>
> Once I was in Fort Lewis, in the officers' mess. I was talking to one of the senior officers there, and he said, "The officers won't use this stuff. It's going to be used by the staff sergeants. We're kind of the executive level here."
>
> He was a bit full of himself. There was a guy from what I think was a special forces unit standing there. I asked him afterwards, "You think officers will be using it?"
>
> "Absolutely. They're going to be some of your biggest users. Let me show you."
>
> And he took me out into the rec room that was part of the officers' mess, and it had this row of video games. And there were all these young officers . . . "NNEEAAAOOO! CLICK CLICK CLICK!" There was Atari, all this stuff, all these fire, fire, bang, bang, bang games.

■ 1984 version of the GRiD laptop, with LCD display, floppy disk drive, and IBM compatibility

Photo
Courtesy of IDEO

The basic design for the GRiD Compass computer stayed the same for more than ten years. There were many different components, with larger displays using plasma and LCD technology, floppy disk drives, and of course all of the electronics. The way in which the screen folded over the keyboard for carrying was one of the innovative features protected by patent, and it was this feature that generated income for GRiD, by licensing the design to other laptop manufacturers such as Toshiba and Sanyo. This geometry remained the dominant design for laptops until the components had shrunk enough to fit them around the keyboard and display, so that a clamshell arrangement became possible without a volume projecting out of the back behind the hinge.

The interaction design advances in this first laptop included the physical interaction of the shape and size, the folding geometry, and achieving a tolerable weight for carrying. The operating system was also very innovative, building on many of the concepts that had come out of Xerox PARC in the seventies. Jeff Hawkins, the founder of Palm and Handspring, worked at GRiD just before the launch of the Compass.

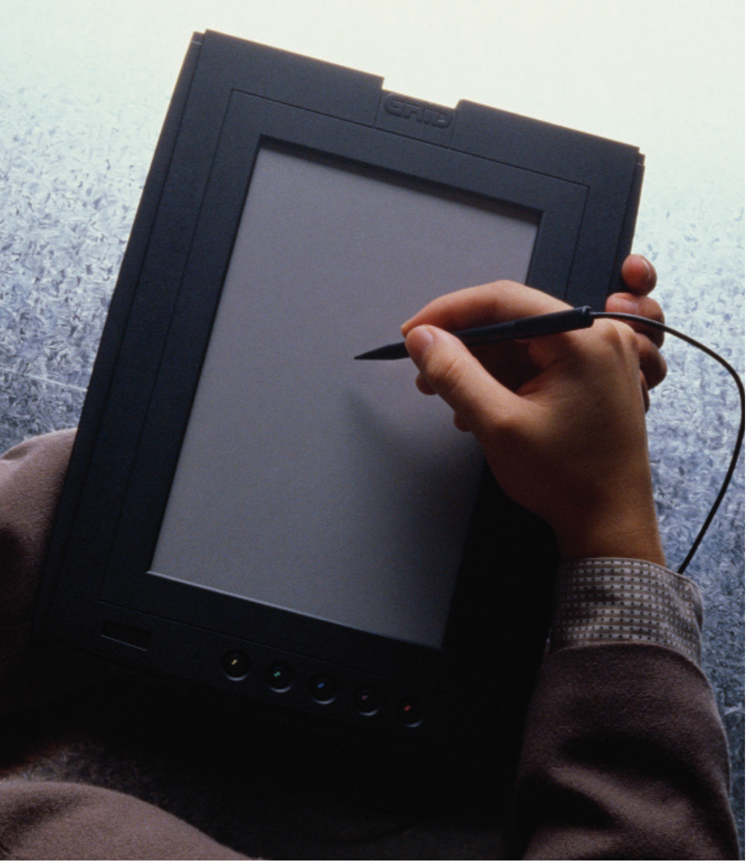
> The GRiD OS was far ahead of anything anyone had conceived of at the time. We had a graphical OS—although without a mouse, but it was a graphical OS—scalable fonts, and a lot of the other concepts that later took hold in the computer industry. It was a true multitasking operating system, it was small, it was fast; it just was leagues ahead of everything else. We had email running on it way before anyone in the business had even heard of email; it had servers; it really had all the technology.

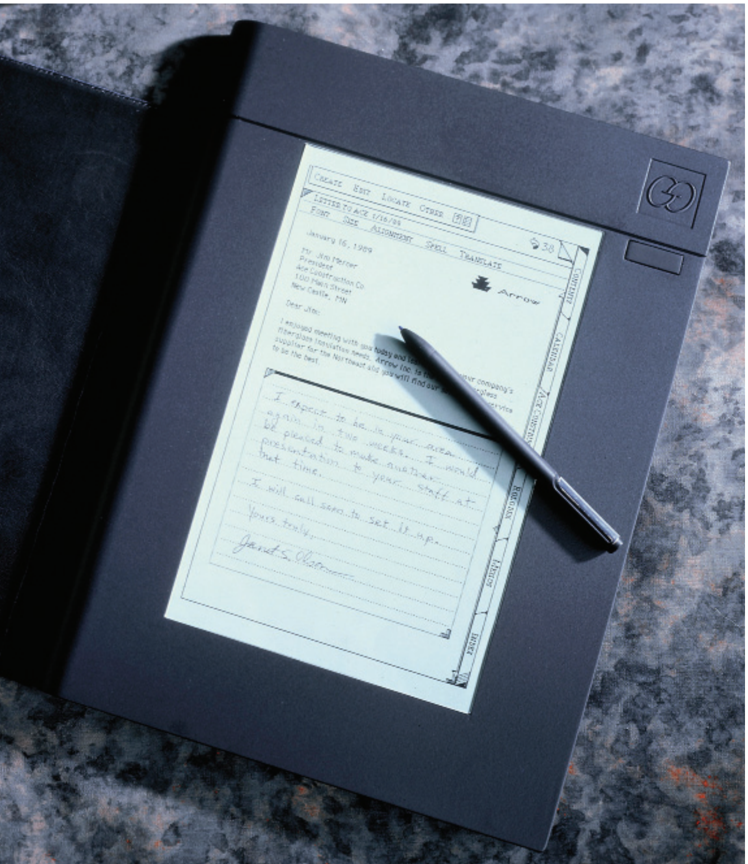
The precedent was set for laptops to follow, but it was Jeff who created GRiDpad, the first tablet PC and the next step toward Palm.

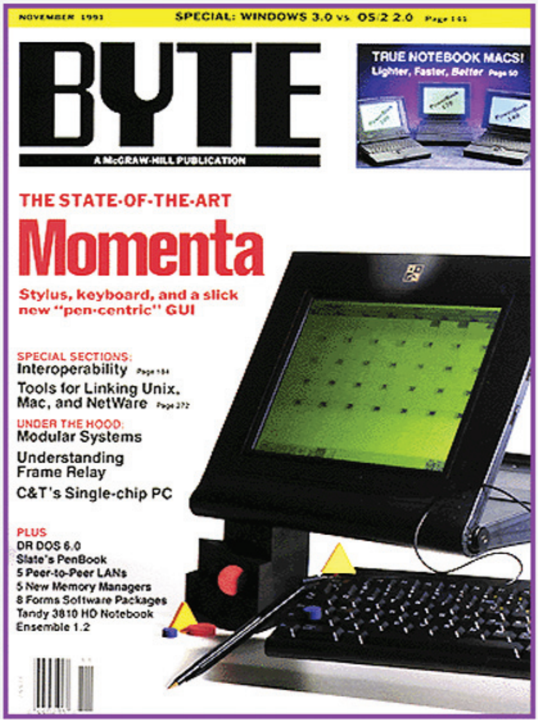
Photo Author

Jeff Hawkins

Jeff Hawkins has two sides to his life: one is designing mobile computers, and the other is working on brain theory. The side that is focused on the design of mobile computers led him to develop the first tablet PC, the GRiDpad, and then to found Palm Computing. At Palm, he was responsible for a series of products from the PalmPilot up to Palm V. He then went on to found Handspring, continuing with the evolution of designs with the series of Visors, and culminating with the Treo platform. The other half of his persona, dedicated to understanding the science of the human brain, has led him to start Numenta, where he is developing a new type of computer memory system modeled after the human neocortex. This technology is based on the ideas in his book *On Intelligence*,[12] which expounds an overall theory of how the brain works, and its implications for the design of computers. He sums it up as, "What makes humans special first and foremost is that we can model the world and we can predict the future; we can imagine the future. How the brain does this is what I detail in the book." This double passion has helped Jeff to clarify his strategies for success as an entrepreneur and designer of handheld computers, while at the same time developing a point of view about neuroscience that has caught the attention of prominent scientists. Jeff went to Intel after studying electrical engineering at Cornell and was attracted to join GRiD Systems in 1982, working as part of the launch team for the Compass computer. He was there for ten years, except for a two-year break to study neuroscience at UC Berkeley.







NOVEMBER 1991 SPECIAL: WINDOWS 3.0 vs. OS/2 2.0 Page 145

BYTE

A McGRAW-HILL PUBLICATION

TRUE NOTEBOOK MACS!
Lighter, Faster, *Better* Page 50

THE STATE-OF-THE-ART

Momenta

Stylus, keyboard, and a slick new "pen-centric" GUI

SPECIAL SECTIONS:
Interoperability Page 184
Tools for Linking Unix, Mac, and NetWare Page 272

UNDER THE HOOD:
Modular Systems
Understanding Frame Relay
C&T's Single-chip PC

PLUS
DR DOS 6.0
Slate's PenBook
5 Peer-to-Peer LANs
5 New Memory Managers
8 Forms Software Packages
Tandy 3810 HD Notebook
Ensemble 1.2

Jeff Hawkins

While I was away at UC Berkeley, I conceived of the idea of doing a tablet computer. I was getting interested in pattern recognition, handwriting recognition. It's easy to put two and two together—a portable computer with stylus interface. It wasn't even my idea, but I saw the potential of it and said, "This is something we can do." I came back to GRiD in 1988 on the premise that I was going to develop the first tablet computer for them. I became the manager of the GRiDpad, the software suite and all the applications that came with it.

Jeff Hawkins, 2004

GRiD

Clockwise from top left

- GRiDpad
- Workslate
- Momenta
- GO

When Jeff Hawkins arrived at GRiD Systems in the spring of 1982, the Compass computer was designed but not yet launched, so he joined as part of the launch team, working on sales and marketing to put together training materials and write demo programs:

> It was very exciting because GRiD was a startup company, and they were inventing the first real laptop, a real innovation at the time. I became involved in figuring out how to sell the product, which was challenging. It was a beautiful machine, but it was very expensive. In 1982 dollars, it was $8,150. I will remember that price forever, as it was an incredible amount of money.
>
> We were trying to sell a laptop to business executives, when at the time no businessperson used a computer or knew how to type. This turned out to be one of the biggest problems we had. The keyboard was something that your assistant or your secretary used; it was associated with word processing. Business people were afraid to use anything with a keyboard: they didn't want to be seen using it;

> they didn't want it in their office, because it made them look like their secretary. One of the largest obstacles, besides the price of the product, was just getting over people's expectations about what a business tool is and what a computer is and how business people use it. When you think back on it, it was amazing that it was difficult to get business people to use a laptop computer.
>
> GRiDtask was a programming language that I developed as a marketing tool, to help people put together demos. I started making it more powerful to give it control of the computer, and to make it do anything you want. Soon our sales people were putting together applications for customers, using it as a sales tool. I was in marketing at the time I was doing this, so no one from engineering was looking over my shoulder. I had unfettered access to do whatever I wanted, and I created a very powerful tool. This became a language, which really became the heart and soul of much of what the company did from a software perspective going forward.

Jeff spent four years working hard with the marketing team at GRiD, struggling to sell the first laptop computer to customers who were not quite ready to pay so much for something that they thought their secretaries should use. The experience served to educate him about the nature of mobile computing, but at the same time he was hankering to learn more about neuroscience, and in 1986 he decided to leave GRiD to become a full-time graduate student in biophysics at UC Berkeley. He remembers the day he resigned:

> John Ellenby was the founder of GRiD. I said, "John, I'm leaving, I'm going to go become an academic, I'm going to Berkeley."
>
> And he was really smart. He said, "Well, maybe you should just take a leave of absence. You might come back. You never know."
>
> I said, "I'm never going to come back, but fine, I'll take a leave of absence."
>
> I left for a couple of years, but even when I was at Berkeley, I continued consulting for GRiD.

Jeff never really escaped the lure of developing mobile computers, or the excitement of working in Silicon Valley startup companies. It was during the two years that he was studying at

Berkeley that he came up with the idea of a tablet computer with a stylus interface.

> While I was away at UC Berkeley, I conceived of the idea of doing a tablet computer. I was getting interested in pattern recognition, handwriting recognition. It's easy to put two and two together: a portable computer with stylus interface. It wasn't even my idea, but I saw the potential of it and said, "This is something we can do." I came back to GRiD in 1988 on the premise that I was going to develop the first tablet computer for them. I became the manager of the GRiDpad, the software suite and all the applications that came with it.

Jeff was not the only individual with entrepreneurial visions of tablet computers. In fact, there was a flurry of new startup ventures at that time. As far back as 1983, Convergent Technologies had come out with a tablet computer called the Workslate[13] that was only 8½ x 11 x 1 inch, but it only had a small display and no stylus input and never found the right applications to make it successful.

By the time Jeff was ready to go back to work, there were several very interesting ventures under way in Silicon Valley. One option for Jeff was to join a startup company called Go:

> The whole tablet computer market was messy. We had the company Go, which took the high road. It had a whole new operating system, high-class hardware, high-class backers, and so on. I went and interviewed with them, because I didn't have to go back to GRiD, as I could have gone anywhere I wanted. I said, "Well, if there's a company formed to do this, I'll go check them out."
>
> I met with Go when there were five people there. I immediately sensed that they had too grandiose plans. They were going to change the world, do all this stuff. Startup companies struggle, and you have to be more focused.

The physical design for Go was elegant,[14] and they made a significant advance in the use of gestures for commands, so that the stylus could be used not just for handwriting and sketching, but also to trigger actions without relying on a pull-down menu structure. Jerry Kaplan chronicled the development of this attempt to start from scratch with a pen-based interface in his book *Startup: A Silicon Valley Adventure Story*,[15] including his loudly voiced challenge to Microsoft that he would dominate the future of computing. He had gone through $75 million in investment funding over a six-year period before he lost control of Go to AT&T.

Momenta was also a high profile startup company, funded by venture capital in the expectation that a general purpose market would emerge for tablet computers. They built a pen-based interface on top of a Windows PC but were unable to stay the pace of improvements in the underlying PC from competitors like Compaq, IBM, and Sony. The design was innovative in the use of the stylus for directional flicking as well as clicking and tapping. They introduced a command structure[16] based on a flying menu that appeared at the tip of the stylus when you tapped, and offered a menu of choices selected by flicking the stylus in one of eight directions.

Apple was another possible opportunity for Jeff, as by this time the research program that led to the Newton was well under way:

> I had heard rumors that Apple was working on something like this; this was before the Newton came out. I talked to them, but at the time they didn't want to tell me what they were up to, so I had no idea that they were developing a completely new operating system. I thought the idea of porting the Mac OS onto a tablet was a more likely direction. In fact, when we were at GRiD, we approached Apple several times, trying to convince them to work with us on doing it, because we felt they had the best software platform for doing this. At that time, they weren't interested. I had an opportunity to pursue it further, but I had two solid options at GRiD and Go.
>
> I decided to go back to GRiD, as I knew that I would be in charge of the whole thing there, and that was more fun for me. GRiDpad was focused on simple form filling. GRiD was an existing company that was trying to survive; we were much more pragmatically focused. We were looking at vertical applications that we knew we could sell. I think GRiD had the most success of any of the new ventures. GRiDpad came out in the fall of '89, a little bit before the other tablet

Apple Newton

> computers. We built a business, I think its peak was around thirty million dollars a year, which was leagues ahead of what anyone else was doing in sales, but that's because we were very focused on vertical markets.

The shakeout in the pen computing and tablet computing area was not long in coming. It is typical in any emerging market that people get ahead of where the real business is. This was a situation where there was such confidence in the potential for pen computing to become the next wave of general-purpose machines that speculation fueled the investment in new ventures, and a lot of money was lost when sales turned out to be limited. It also got out of control because there was a big rivalry between Go and Microsoft, which stepped up the level of investment in both companies, but the business just wasn't there. Jeff believes that the problem was that these products were trying to replicate paper, putting them in competition with paper and books, a situation he has always tried to avoid:

> I think paper is just this wonderful medium; it's been honed for a thousand years. It's really great! To try to do what you do with paper, with just drawing—line width, sketching, and so on—it's very hard to get a good experience. Where the tablet type of computer really shines is where you're not trying to capture the paperness of paper, but you're trying to get the electronic or the back end of it. Form filling is a great example, because actually forms are pretty hard to do on paper. You never have the right space, it's hard to know where to put things, there's not enough room for instructions. So, there's an example where an electronic version of a paper equivalent would be better. But in terms of the general idea that I'm going to sketch, draw and have a free-flowing paperlike experience—I'm skeptical about that.
>
> Similarly with electronic books. Paper in books is just wonderful. It's a beautiful medium. It's superior in many ways to any sort of electronic form, especially if you're trying to create something in a similar form factor. Also for the readability of text. Paper is amazingly good for different light angles and brightness and so on. It's hard to get that on today's digital technology.

GRiD

In 1992, after building GRiDpad into a successful business, Jeff decided to see if he could develop a product that offered the best of both the laptop and tablet, in a project called the GRiD Convertible:

> Instead of the normal laptop, where the display hinges down and covers the keyboard with the display hidden, we had a very clever design—well, I thought it was clever—where the display was up, and it had this double hinge and it came back; it had a nice mechanism to it. We brought that to market, but what we found out is that people don't really want a laptop and a tablet combination. It wasn't that the technology wasn't ready, it just turned out that fundamentally it didn't seem that this was something that people really wanted to do.[17]

This was a surprising but hard lesson, which stayed with Jeff ever since, making him skeptical about the efforts that Microsoft is making in the new millennium to boost the tablet PC. At this point we step aside and turn to Bert Keely, the "architect of tablet PCs and mobility" at Microsoft, to understand the ideas and interactive advantages that Microsoft is betting will make the tablet PC come into its own.

■ GRiD Convertible

Photo
Courtesy of IDEO

Photo Nicolas Zurcher

Bert Keely

“The essence of what I’m trying to bring to the personal computer, you should probably think of as a ‘tablet mode,’ that is, the easy option of using the computer with one hand, even when you’re holding it with the other. While speech, touch, and buttons will all have a role, the pen is the most exciting tool with which to do that in the current generation of PCs.” When Bert Keely graduated from Stanford with an engineering degree, he went to a startup called Convergent Technologies, where he spent eight years developing desktop work stations. He first latched on to the concept of a handheld computer in 1988, when he joined John Ellenby to develop rugged devices for the police, fire service, and military markets at Agilis,[18] but the company was short-lived. Next he joined Silicon Graphics to develop work stations for design and scientific applications. It was there that he built an advanced prototype for a magazine page display with a powerful workstation behind it; the information, he thought, should be at the center of manual interactions. Silicon Graphics decided against taking this concept to market but gave Bert the opportunity to shop it around, leading him to a dialog with Bill Gates at Microsoft. He joined Microsoft in 1998, working first on software for electronic books, and then on a skunkworks tablet PC project, designed to help people take notes silently and intuitively in face-to-face meetings; this was launched as a service pack on top of Windows in 2002. Since then, Bert has been leading the efforts at Microsoft to improve the design of the tablet PC, earning the title of “architect of tablet PCs and mobility.”

acer
TravelMate C200
OPEN
Windows Media Player 11
Share Your Library
You can share digital media files with other computers and devices on your home network, allowing you to enjoy music, pictures and video anywhere in your home.
Remind me to set up sharing
Recycle Bin
AcerC200
Link to Desktop
Sticky Notes
Snipping Tool
How to Share Feedback
IDEo - (Read-Only) - Microsoft Office O
IDEO
100 Forest
Alma
University
Windows Media Pla...
Windows Defender
IDEo - (Read-Only) ...

Bert Keely

The breakthrough will come when mainstream users find themselves thinking, "Oh, there are lots of situations in which I just use tablet mode, because of the physical freedom I get. I don't mind the fact that a keyboard can enter more words per minute than handwriting can, as I prefer tablet mode most of the time!" With the second service pack for Windows XP, some users are already there.

Bert Keely, October 2004

Displays

■ Acer TravelMate C200 Tablet PC

WHEN BERT KEELY arrived at Microsoft in 1998, he brought with him a wealth of knowledge and experience of displays and high-resolution graphics, so the first question he was asked to address was,"What can we do to improve the visual characteristics of on-screen reading, to make the display more like paper?"

He suggested a project to see if they could take advantage of all those little subpixels on the display and went on to develop ClearType, a program that takes advantage of the red, green, and blue subpixels, to smooth the fonts and create the best word shapes on screen. He explains how this works:

> Anti-aliasing is the process of saying, "Okay, I don't have an edge that directly represents the edge of the character, so what I'm going to do is kind of smear the edge I do have. I'm not going to have white on one side and black on the other side of that edge, I'm going to have grey on both sides of that edge." ClearType starts by using the edges within the pixel, the edge between the red and the green

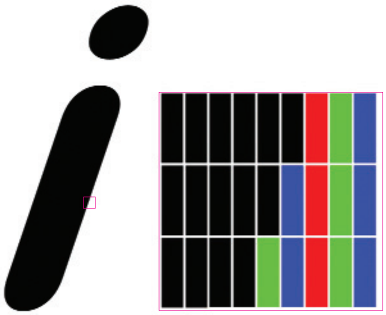
stripe, and the edge between the green and the blue stripe, and adds anti-aliasing within the pixel.

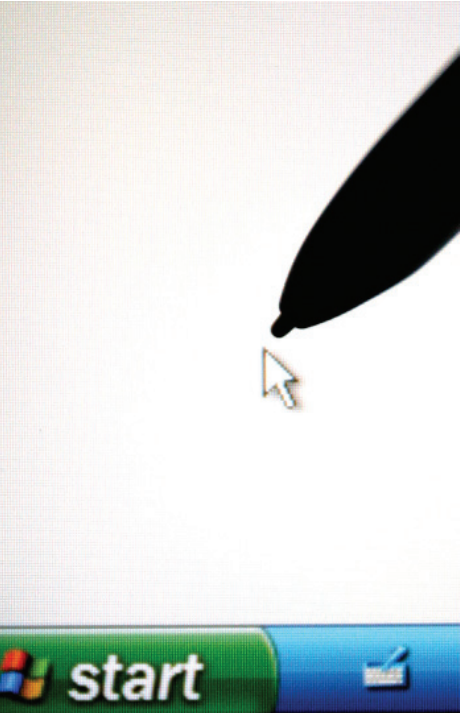
As the eye has very different sensitivity to red, green, and blue, it turns out that it's not a simple matter of smearing; we had to adjust the luminance values. Then because the human eye is much more sensitive to luminance edges than it is to color edges, we were able to take advantage of the eye's willingness to accept some color error in order to create much sharper fonts.

A challenge for the design of displays for pen-based computers is to precisely align the sensed location of the pen with the actual location of the pen. The ideal would be to eliminate the displayed cursor and just have the pen tip represent itself, but this is hard to achieve.

More important than parallax, what you actually see on most tablet PCs is just a pure XY misalignment between the position of the pen tip and the coordinate the digitizer is reporting, either because of a lack of linearity or a lack of calibration. People commonly call this parallax, but it's not a depth issue, it's an XY error issue. Actual parallax, which means a perception of depth difference between where the tip of the pen is and the image plane, is not as big a factor. Notice when you're looking at an angle down into a swimming pool, the foreshortening of the image actually translates it nearer to your line of sight through the surface.

What you want is to have the digitizer be accurate enough to really sense the location of pen tip and then let the user just think in terms of positioning it. When they see an onscreen object, they should be able to point at it directly with the pen tip, the way they did when they were three or four years old, learning how to draw with a crayon. They shouldn't have to worry about positioning some direct but slightly indirect cursor.





- Close-up of pixels
- XY error between pen tip and cursor

Portability

A STYLUS INPUT should be designed to let you interact with your computer with only one hand. This in turn promotes portability, as you can be holding the computer with the other hand while you stand or walk around, or you can be sitting at a table and have your hands on something else, like some food or some documents that you are looking at, and use the computer at the same time.

The cell phone is something you carry with you all the time—WYATT, as Bert says. The personal computer can become something that you want to have with you at least most of the time, because it carries your whole digital world. Bert sees PCs becoming the second device for each person in a family to carry with them, so that the phone and the PC become digital companions:

> Your family pictures, your music, your calendar, your contacts, your recent email threads, the documents that you're reading, other things that you might have looked at recently that you want to get back to, every Web site you've seen since you bought that computer, all of this stuff; having your whole digital world on hand is something that the PC lets you do, because it's fundamentally visual. You find yourself, when you go to dinner, a meeting, or an outing in a car, thinking, "Well, if I've got every street on the United States on that thing, and I'm comfortable with the user interface because I've figured out how to use it, that's going to be the handiest way to carry maps, and all my visual information.

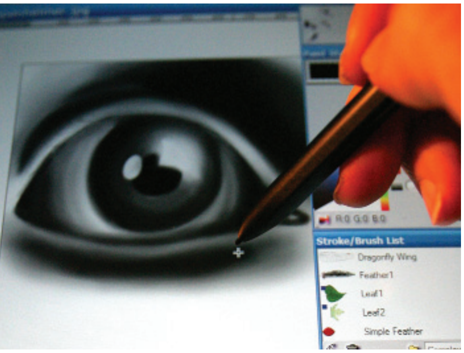
Standing use ■

Pen and Paper

THE METAPHORS OF desktop and window were originally helpful to the novice users of PCs, because they made the behavior of objects in the abstract digital world relate to the familiar experience of the physical world. The personal computer has evolved a lot since then, and one can argue that the design has outgrown the metaphors. The actual behaviors of the electronic objects have more meaning to the user than the simulation of a physical world that now seems archaic; many children start playing with computers when they are two or three years old, probably before they have ever seen a file folder. Bert believes that the pen-based computer will grow up in a similar way, starting by trying to simulate pen and paper, and then evolving unique characteristics that have special advantages in the digital context.

The main uncertainty is whether it will take off in the first place, as there has to be an immediately recognizable value for people to make the leap from the laptop to the tablet. Jeff Hawkins is skeptical about this potential, in spite of the fact that he developed the first successful tablet, but he stayed clear of simulating paper by concentrating on form-filling applications. Bert Keely is much more ambitious and hopeful, and he bases his optimism on the unique value that pen-based computing offers. He sees the tablet supporting single-handed interaction, and hence operation when standing, or in social situations. He also believes that the ability to capture thoughts in a free-form fashion is unsurpassed with the pen. You can sketch and enter text with one hand with a fluency that you have been developing from childhood. It is impossible to do that with just a keyboard, or with voice input, and it is difficult to do it with a combination of keyboard and mouse.

Bert was experimenting with pen-based inputs when he was still working at Silicon Graphics:



■ Drawing

> We were focusing on how to make the pen and the finger great pointing devices, and great devices for capturing thoughts. That was because essentially we wanted to work directly with the display, and

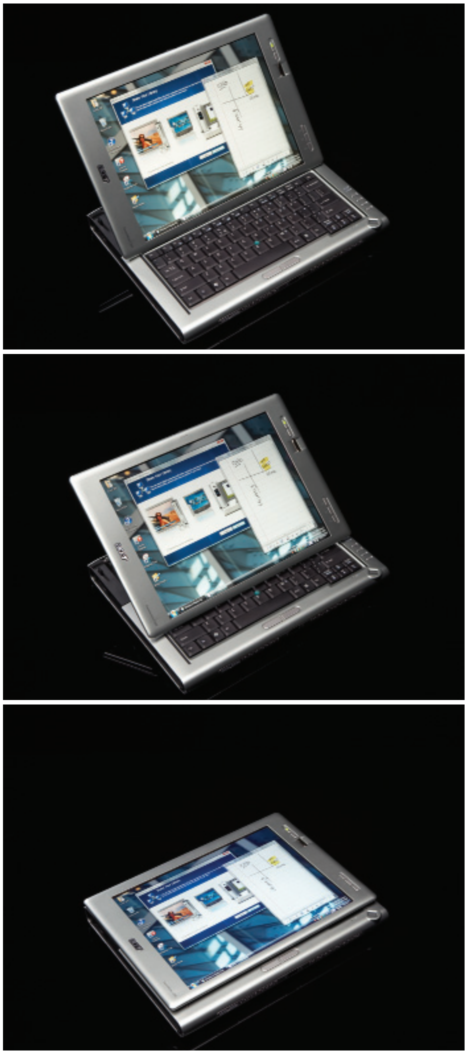
we knew that we needed a pointing device. The graphical user interface had clearly proven itself. We knew that the pen was excellent for capturing thoughts, and the finger was probably the most natural and most convenient thing to use for just reaching out and touching some piece of information.

The classic thought with regard to the electronic pen is that you want to make it look and feel and behave as much like a pen on paper as you possibly can. Now, the reality is that there are many different types of pens that behave very differently on paper, and pencils of course, and there are also many different types of paper. So, there's quite a range of "pen on paper feels" that people can become accustomed to.

In the end, I think it's going to remain a user preference. Some people love the feel of a fountain pen, and they even like the sound that a fountain pen makes. The feel of a fountain pen is literally a piece of steel scratching across fiber; that's a very rough, very high-friction feel. The essence of a ballpoint pen is that you're rolling almost frictionless, but if you press hard, you get the tooth of the shaft that's actually trapping that ball, so you get a little bit of that scratch. The feeling of a pencil, of course, is one of graphite being deposited on the paper. One thing about a pencil is that the static friction is low, but there's lots of sliding friction, because you're actually pulling that pencil apart. So, what is the static friction? What's the sliding friction? What's the ratio between those two? What is the sound that the system makes? All of those are factors in "emulating" pen on paper.

Today we have a few coatings that can be placed on glass or on plastic that really do a good job of establishing the kind of "stiction" that makes a person feel comfortable tapping, and makes them feel more or less comfortable writing. We still have lots of experiments going on with regard to the kind of sliding friction that's best. Once a person becomes accustomed to less friction and to the efficiency of movement that the slipperiness brings, they tend to like it and find that going back to paper starts to feel a little more tedious.

With a pen-based user interface you have the opportunity to leverage motor skills, which go far beyond the motor skills of moving a bar of soap around, which is essentially what it's like to use a mouse. When we design for the pen and finger, we are trying to leverage the fundamentals of being able to do expressive things with your hands.



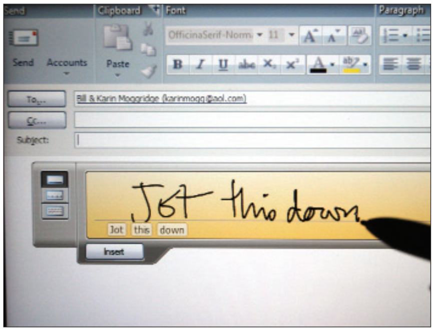
From keyboard to stylus ■

Successful handwriting recognition may be a basic requirement for the tablet PC market to really take off. Microsoft is making a huge investment in what they call “Ink Analysis,” or the recognition of shapes within ink marks on paper or tablet. They are also studying the context in which particular strokes are being created, looking at the text to the left and the text to the right, and using those as part of the decision of what to elevate in the list of candidates. Analysis of the vocabulary of the individual user can also narrow the possibilities effectively, allowing the system to make better guesses about what each scrawl might represent.

Bert thinks that Microsoft is making good progress in handwriting recognition, but his optimism about the future of tablet computers is also based on the combinatorial advantages of using a stylus:

> Something that you only find out when you’re using a pen as a text input device and also using it as a pointing device is that the interleaving of pointing and writing happens so fast that you really are able to compose as quickly with a pen as you can with a keyboard. This isn’t a function of handwriting recognition being an equal number of words per minute. This is a function of handwriting recognition, plus pointing, plus correction, plus placing your cursor, selecting, dragging text around, reformatting, and undoing. You’ve got this combination of things you’re trying to do, some of which is text input and some of which is pointing. When you can do that in a fraction of a second back and forth between the activities, it’s really astounding.

Using handwriting or voice recognition software gets very frustrating if there is even a very small proportion of errors, as it is annoying to have the flow of your consciousness interrupted by the need to make corrections. Many people attribute the failure of the Apple Newton to the lack of reliability of their handwriting recognition. Artificial intelligence (AI) promised to solve these problems but failed badly in the twentieth century, leaving the keyboard, and hence the laptop, in a dominant position. Jeff Hawkins had been studying how the brain works in



To... Bill & Karin Moggridge (karinmogg@aol.com)
Cc...
Subject:

Jot this down

Jot this down

Handwriting recognition

parallel with his work on tablets and he thinks that it was his knowledge of "consciousness" in a literal neuroscientific sense that led him to take a lateral approach to handwriting recognition. We now return to him, as he describes the development of the PalmPilot and Graffiti handwriting recognition software.

PalmPilot
3Com
Address
Calc
Date Book
Expense
Giraffe
HotSync
Mail
Memo Pad
Memory
Prefs
Security
To Do List
10:41 am Battery
abc
123

Palm Computing

3Com PalmPilot

Photo
Courtesy of 3Com

While at GRiD, Jeff Hawkins learned how to design the software and hardware of laptops and tablets and how to understand the business issues that limited the scope of the market. He began to yearn for the opportunity to reach more people than was possible in the vertical market segment where GRiD was operating—selling business applications to other businesses. His intuition told him that the future of portable computers was likely to be for smaller devices that appealed to consumers:

> When I was at GRiD, we made laptops and we made the GRiDpad. When we were still doing the GRiDpad, it struck me that a lot of people loved the product. I had numerous people come up to me who were using it for some vertical application who said, "You know, if you could only make this thing smaller and I could put my personal data on it, it would be really clever."
>
> I heard this a few times, and said, "Well, this must mean something."
>
> I realized that the idea that people wanted a small computer to carry around with them was very powerful. It struck me that, in the future, everyone would be carrying something in their pocket; Moore's law would just lead to that ability. I didn't know what it would look

> like yet, but I concluded that this was inevitable. People will have small electronic devices in their pockets, or on their belts. They will have storage and will be little computers of some sort. I decided this was a great thing to work on—not just fun, but also important.
>
> I started seeing desktop computers as clunky things. What a big monstrous machine on your desk! That can't last forever. And as soon as you have five of them, you have to have an IT professional to maintain them—this was impossible. I said, "This is not a computer for the world. This is not going to work in a world of billions of people. You can't have these big, complex, power-hungry boxes in everybody's life."
>
> So, the idea that something portable would be the main interface for the world just struck me as an inevitable consequence of the evolution of technology.

Jeff realized that GRiD was not the right company to support the birth of a consumer electronics company, so he decided to leave and founded his own startup.

The idea of smaller computing devices was taking hold by this time, even if there had been no commercial successes so far. Back in 1987, Apple had created a vision of computing in a video called the "Knowledge Navigator"[19] that featured a professor being helped by a humanoid avatar with a much-more-than-feasible level of artificial intelligence. The video was created by a design team who let their imaginations run wild, but it was widely publicized and very influential, causing the name of PDA (personal digital assistant) to gain popularity as a generic label for the next generation of smaller and smarter machines.

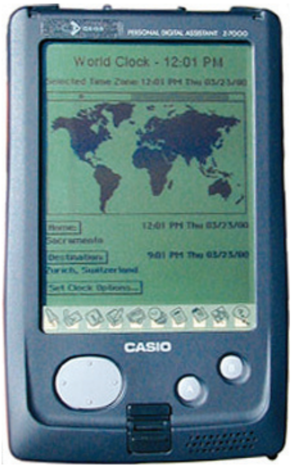
Jeff was searching for a name that would fit within the overall context of the PDA but would be strongly differentiated from this connotation of artificial intelligence. He came up with the name Palm Computing, but at first it was controversial:

> The "Palm" part was easily accepted, as it was a friendly word, it was international, and of course it could apply to your palm. The controversial thing in "Palm Computing" was that I called it "Computing" as opposed to "Palm Software." Software was hot when we were starting the company in late 1991. In the venture capital world, everyone was building software; no one would build hardware.

> I was being pushed to call it "Palm Software," as we were initially a software company, but I felt that as we were trying to build the next generation of computing, none of us knew where that would take us. I consciously said, "I want to call it 'Computing,' because that incorporates anything we might want to do in this space." "Palm Software" would be more limiting. My colleagues and backers didn't like it; they were scared of the idea that sometime we might do something other than software.

Palm Computing started by writing software for the PDAs that were being developed by large companies with plenty of resources to invest in R&D, partnering first with Casio to develop the Zoomer, and then with a consortium of other companies, including Sharp. They found that they were being forced into compromises in the design of the software by first one partner and then another, so Jeff was getting increasingly frustrated. He remembers a particular day when he was bemoaning the situation to one his board members:

> I said, "We don't have much money, and it doesn't look like anyone's going to build PDAs anymore," because so far everyone had failed with their PDAs.
>
> So he asked me, "Do you know what product you should be building?"
>
> I hadn't really sat down and thought exactly what I would do if I started from scratch, but I kind of lied and said, "Yeah, I know what we should be building! We have to do the whole thing. We have to do the hardware, the software, the operating system, everything. I have to do it from scratch."
>
> He said, "Well, that's what you should do then!"
>
> I said, "All right, great."
>
> Within a day and a half, I had a design for what ultimately became the PalmPilot.



Casio Zoomer

Design Criteria for the PalmPilot

By this time in 1994, Jeff had learned a lot from working on Zoomer (Palm's failed product with Casio) and by examining Apple's Newton and the General Magic products, so he was ready to make some bold decisions. In that first day and a half, he came up with four design criteria to define his premise; size, price, synchronization, and speed.

Size

He went home after the defining conversation and made a wooden model that same evening, cutting it down until he felt that it would fit conveniently in the pocket of a shirt or jacket. He then wrapped the block of wood in paper and presented it to his design team.

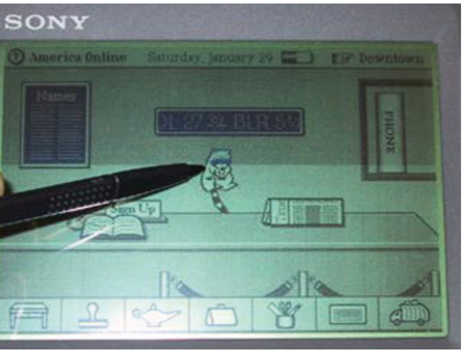
Price

He picked a price point of $299, as he believed that was the most that people would pay for a consumer product with broad appeal. It seemed very hard to achieve at the time.

Synchronization

Another goal was to have synchronization with your desktop computer built into the design. Up to that time, everyone thought that synchronization should be optional, but Jeff realized that it was the most significant advantage that could be offered for a pocket-sized device:

> Synchronization wasn't an add-on—it was a core application! Credit goes to Walt Mossberg of the *Wall Street Journal* for that. He had written a critique of some of our earlier products, saying, "This is crazy! Why should you be selling this add-on that everyone needs? If everyone needs it, include it in the product. He was absolutely right, so we said we were going to design around synchronization.



- Sony General Magic screen detail
- Sony General Magic product

Speed

The last attribute was that it had to be fast. Jeff realized that he was competing with paper, not computers, as he was trying to get people to replace their paper systems, not their digital systems. He used the metric that you had to be able to do things just as fast with the new design as if you were using a paper-based personal organizer. This was a difficult challenge as the price point meant that they could not put a fast processor in the first version of the product. Jeff talks about the implications:

> The software guys came back and said, "We can't do all that stuff you want to do in software with that slow a processor."
>
> I said, "Sure you can, because when we were at GRiD, we had a more sophisticated operating system which ran on a slower processor, and it was fast. So, if we could do it back then, we can do it now."
>
> Sure enough, they were able to do it. They made a product that was very fast. One of my design goals was that there would be no "wait cursors." I said, "Why should there be wait cursors? That's a mistake. We don't want those. That's a crutch for a bad design. Ban them!"
>
> I also insisted that we would have no error dialogues, especially the ones that ask a question of the user, where the user has no choice, like, "You have an error in the fla-fla-flop file, what do you want to do? Continue or abort?" Why ask the user a question like that? Just do something, don't even ask them.

Those four design criteria solved a lot of the problems that were prevalent on personal computers and PDAs at the time, and turned out to be a defining formula for success.

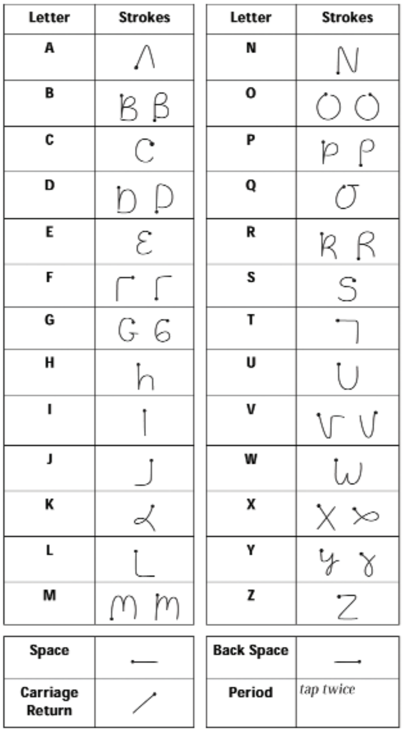
Graffiti

While Jeff was studying brains at UC Berkeley, he went to a conference about neural networks and came across a company that built handwriting recognition software. They were offering the package for sale for a million dollars, which made Jeff think it might be worth developing his own version of handwriting recognition software. Two days after the conference, he had a working prototype of an approach that looked promising, starting him down the path of designing traditional handwriting recognition.

When he returned to GRiD, he made his approach work well enough for forms, with numbers and some limited text entry, but it never worked well for anything sophisticated. When he started Palm Computing, there was a lot of pressure to produce handwriting recognition software for the early Palm products, like the Zoomer, but he resisted, saying, "I don't think it's going to work. What I want to do instead is focus on ink."

This led to work on capturing and resizing the shapes of handwriting and sketching, but without recognition. Ultimately he realized that the value of this was limited, and that he would have to come up with a way of entering text into a handheld computer with character recognition. In 1994 someone sent him an email with the title of a paper by David Goldberg and Cate Richardson from Xerox PARC, "Touch-Typing with a Stylus."[20] Jeff describes the epiphany:

> That was the spark that I needed. I didn't even read the paper. I just saw the title, which inspired me. It made me think of the analogy of the keyboard, which is hard to learn; to learn to type takes months, but everyone does it. The accepted paradigm at that time was that computers should adapt to the users, not the other way around, and I said, "Why is that? Well, people don't mind using a tool as long as it helps them perform the task. I think they like to learn. They pride themselves on learning how to use tools, as long as the tool is consistent, it's learnable, it has a good model, and it's reliable." So, the idea of Graffiti was to say, "Okay, throw away traditional handwriting recognition, and then do something which is reliable and works."



Letter	Strokes	Letter	Strokes
A		N	
B		O	
C		P	
D		Q	
E		R	
F		S	
G		T	
H		U	
I		V	
J		W	
K		X	
L		Y	
M		Z	
Space		Back Space	
Carriage Return		Period	tap twice

Graffiti alphabet

> It was easy for me to see this because up until this time, the old handwriting recognition software I had created always worked best with my handwriting, because I was the guy who created it. One of the industry analysts quipped, “If you write like Jeff, it works all the time!”
>
> I tried to generalize and said, “What if we just told people to write a particular way?” What if we said, “Write your *e* this way and your *f* this way and your *g* this way!” It would reduce the problem significantly. That was the genesis of Graffiti. We just came up with a simplified system that was fast and reliable, based on the way I write an upper case character set; then we tweaked it from there.
>
> I don’t think I would have come up with this idea if I didn’t understand some of the stuff about brains already. I knew that brains want a consistent model, and they want to be able to predict what’s going to happen. The problem with traditional handwriting recognition was, when you wrote something and it didn’t work, you had no way of knowing why. You wanted a system where you could say, “All right. If it fails, I know why; it was my mistake, I’ll fix it. I can change my behavior.” That gives you a predictable relationship between behavior of the machine and your own behavior.

When Jeff finally did read the PARC paper, he was impressed by the research, but he realized that the goal of the authors was to create fast text entry, whereas his goal was to create reliable text entry that would be easy to learn. He focused on trying to make the pen strokes as much like regular handwriting as possible, keeping the characters very close to standard letterforms. The PARC approach was designed to maximize speed, so they chose the simplest strokes for the most frequently used characters. The *e* was a dot, because that was the most common letter, and the *s* and the *t* were single strokes, with more complex strokes reserved for the least common characters.

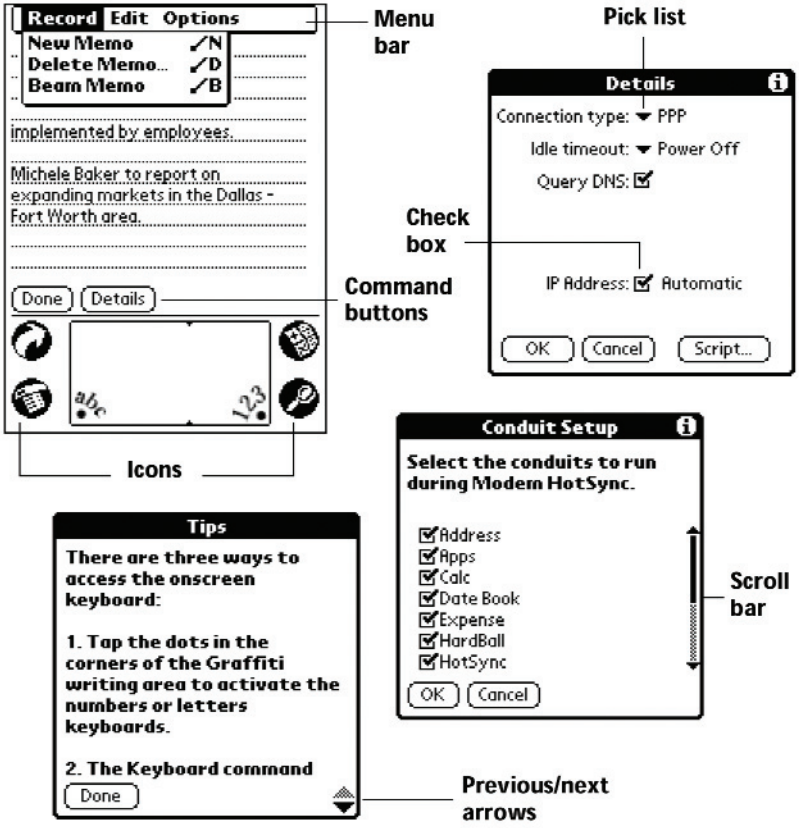
The combination of Graffiti and Jeff’s four design criteria for the PalmPilot were the keys to a good product, but the dramatic success that they enjoyed was also due to the unique interaction design. For this, Jeff relied on Rob Haitani, who had worked with him from the very first days of Palm Computing.

Photo Author

Rob Haitani

Rob Haitani was responsible for designing the user interface of the operating system for Palm, as well as all the original applications. He is modest about his abilities, claiming to be an "applied practitioner," as he has no formal training in interaction design. One morning the president of Palm, Donna Dubinsky,[21] asked him if he had ever done this before. He wondered if she meant whether he had designed a user interface, but as he had never even managed a software project before coming to Palm he said, "Uh, no." Before joining Palm he worked at Sony in Japan for several years, in the camcorder group. There he was surrounded by people who were dedicated to miniaturization, which proved to be useful experience for Palm. After Sony he worked on digital video cards for Macintoshes, and when he joined Palm, he was asked to work on a project to answer the question, "What can we do about making an organizer work better?" It was something he knew nothing about from the product side, but from the customer side he could very easily pick up one of those products and realize that there was something wrong. Rob insists on putting people first and brings a combination of common sense and ruthless analysis to any problem that he is confronted with, never forgetting the end user. He stayed with the team when they spun out of Palm to start Handspring and went on to lead the interaction design team for all of the Handspring products. In 2004 Handspring and Palm came together again to form PalmOne, with PalmSource as a separate company for the Palm OS operating system.

Elements of the organizer interface



implemented by employees.

Michele Baker to report on expanding markets in the Dallas - Fort Worth area.

Connection type: ▾ PPP

Idle timeout: ▾ Power Off

Query DNS: ☑

IP Address: ☑ Automatic

Select the conduits to run during Modem HotSync.

☑ Address
☑ Apps
☑ Calc
☑ Date Book
☑ Expense
☑ HardBall
☑ HotSync

There are three ways to access the onscreen keyboard:

1. Tap the dots in the corners of the Graffiti writing area to activate the numbers or letters keyboards.

2. The Keyboard command

Menu bar A set of commands that are specific to the application. Not all applications have a menu bar.

Command buttons Tap a button to perform a command. Command buttons appear in dialog boxes and at the bottom of application screens.

Icons Tap the icons to open applications, menus, Calculator, and to find text anywhere in your data.

abc With the cursor in an input field, tap the dot to activate the alphabetic keyboard.

123 With the cursor in an input field, tap the dot to activate the numeric keyboard.

Rob Haitani

I developed "Zen Riddles" to articulate points in a way people would remember. The most core point was the riddle of "How do you fit a mountain into a teacup?" When I would present it, people would say, "Well, you have to shrink the mountain." If that's what you're thinking, then you're still thinking more is better—the PC thinking. What you really have to think is, "Why do you want to put a mountain in a teacup to begin with?" Ask the question about what really matters before you decide how to act.

Rob Haitani, 2001[22]

Interaction Design for Palm OS

■ A page from the manual describing the elements of the organizer interface

Rob Haitani was developing the user interface of the Palm OS in 1995, when Newton had been out for about two years and it was clear that the initial excitement about the design was not going to be enough to make it a hit in the market. The conventional wisdom at the time was that products such as the Newton were not succeeding because they did not have enough functionality; if you are going to spend $700 on something it should do more, and you should pack it with features. Jeff Hawkins had articulated the goal of making a really simple product and defined his four design criteria of size, price, synchronization, and speed, and he was always there for the meetings when the design team was formulating their ideas for the new operating system. Rob led the team in building a consensus for simplicity and pushing back against the pressure for feature-laden functionality.

They tested Jeff's criteria in some customer research with focus groups, showing a form factor model to the participants, with the user interface on a Macintosh screen to give them a sense of what the interaction would be like.

First they showed the organizer on its own and said, "What do you think?"

People said, "That's interesting, the form factor is nice, it seems to be easy to use. How would I get the data into my PC?"

In the second stage they showed the cradle and said that the organizer could be synchronized with the PC and got a very enthusiastic response. In the third stage they said it could do wireless email, but very few people were interested:

> Wireless email was really hot back then. We're doing that now, but at that time it was not ready for prime time. We were asking people, "Would you like wireless email?"
>
> From our Silicon Valley bleeding edge mentality we thought that people would just be falling over each other to get it, but they were just shaking their heads. I remember very clearly one woman saying, "Well, I get about three emails a day. I log in the morning and I look at them; why would I want these in a portable device?"
>
> She didn't even understand what the concept was. That's when you really sit back and say, "Wait a moment, we've got to remember who our customer is."
>
> We're not the customers. We're geeks. We're the Silicon Valley propeller heads and not the normal people who will be using our products. Focus groups are most valuable when you have something that you can show not only the potential, but also the limitations, of the product. I remember thinking when we did our first focus groups with the PalmPilot, "Boy, the Newton must have just flown through these. Imagine a hand-held product, and it recognizes your handwriting. Would you like it?"
>
> The answer would have been, "Oh wow, that's great, I love it!"
>
> If you had a real prototype, most of the time it wouldn't have worked and it would have been really slow.

HotSync in cradle

The limits on size and cost that Jeff demanded forced Rob into using a monochrome display with only 160 x 160 pixels, posing a challenge for the interaction design. The 320 x 240 pixel display on the Zoomer had seemed incredibly difficult to manage, and yet Rob was now faced with the task of reducing everything on the screen to an even smaller size and still conveying the same amount of information.

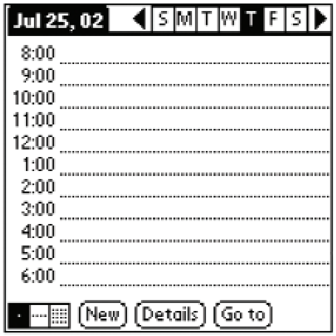
> Jeff defined the resolution and said, “Can we design an operating system on this resolution?” and I said, “Sure, of course, no problem!” and afterwards I thought, “Oh dear, it’s going to be difficult to make this work.”
>
> You have to focus on what really needs to be conveyed on the screen. Ultimately this was a positive factor driving the user experience, because it forces you to be disciplined. For example, on this size of screen, instead of having a whole row of buttons, you can basically fit four buttons on the bottom of the screen, and the question is, “How can you decide which four buttons are on the screen?” You can either try to cram ten buttons in there and have this horribly cluttered interface, or you can decide which those four buttons are.

The calendar application was difficult on such a small screen. In the focus groups, people insisted on the importance of seeing a whole day, but even Newton, with its much larger display, only showed the day from 8:00 a.m. to 3:00 p.m. By using a font with a capital letter height of only seven pixels, Rob managed to create a screen that contained a row for each hour in a ten-hour day and still have room at the top and bottom for menu and controls. The screen was made to look higher by positioning the stylus pad for Graffiti handwriting recognition at the base of the display in an included rectangle.

Rob was starting to revel in a reductionist philosophy, and found that it allowed him to define the interaction design for the whole Palm operating system. After a while the ideas became ingrained in the attitudes of the original team of designers, who started to intuitively design to these four guidelines:

Less is more.
Avoid adding features.
Strive for fewer steps.
Simplicity is better than complexity.

That was the right approach to take in 1995, when the main problem with products was that they were trying to do too much. Products were not only too complicated, but developers were going in the wrong direction; they were trying to add more functionality and more features.



Jul 25, 02 S M T W T F S
8:00
9:00
10:00
11:00
12:00
1:00
2:00
3:00
4:00
5:00
6:00

Four buttons and Graffiti pad ■
Calendar application in 160 x 160 pixels ■

The "Zen of Palm"

Once the initial products were launched, Rob developed ways of explaining his design philosophy that he called the "Zen of Palm":

> The "Zen of Palm" was not articulated as a design philosophy until the third-generation Palm OS. The concept was developed because we were growing as a company, and there were more and more people involved in the design; I got tired of saying over and over again, "No, no, we have to keep it simple."
>
> More people would join the company and want to add more features, saying, "Well, Microsoft is adding more features. . . ."
>
> It was developed as a way to teach people the philosophies that the core team knew instinctively. I was trying to articulate the ideas in a way that people could walk away and remember. I was sitting in my office at ten o'clock at night, preparing a presentation for the next morning, and I was trying to think of a funny way of capturing this. People had referred to Palm products as Zenlike, so I thought, "Okay, let's play on that theme!"
>
> I developed "Zen Riddles" that would try to articulate points in the way people would remember.

Rob put some presentation material together to show at developers' conferences and within the company. He created a chart that represented all of the features and how often you would use them. It had a few features at the top that you would use very frequently, and then it would dramatically drop off, forming a sort of bell curve. He then posed the question, "Which features do you put access to right on the screen?" He then showed the bell curve with the three of four most frequently used features at the top marked for inclusion on the screen, and all of the others forced below a menu or control. He illustrated this principle with an analogy:

> If you think about the way your desk is organized, you have some things on your desk, and some things in your drawers. Why is that? The things on top of your desk are there because you want to access them very frequently. Imagine if you had your mouse, or your phone,

> or something that you access very frequently, and you put it in your drawer. Every time you wanted to use it, you'd have to pull it out of your drawer. It might be only one more step, and it might be only one second, but it would really drive you crazy over the course of a day.
>
> Think about your stapler and staple remover. My stapler is on top of my desk, and my staple remover is in my drawer. The reason is that I staple papers more frequently than I unstaple them. You can argue that architecturally speaking the stapler and staple remover are equivalent and therefore should be in the same place. If you look at it intuitively and ask what you do more frequently, some of these decisions just naturally bubble up to the top. It all depends on understanding your customers, but not on a very complex level. It is not rocket science to suggest that you would be more likely to enter a new phone number than to delete one.

He emphasized the importance of instant access to data with the analogy of a watch, which shows you the time immediately. If you had to wait three seconds before you could see the time or had to a press a button to see the time, you would choose a different watch. If you need two or three steps to gain access to something that you only use occasionally, it is acceptable. This raises the question of how to hide the less frequently accessed functions? Do you use menus or modes?

> One of the first answers to the problem of not being able to have a button for every function was, "What if we were modal about the buttons? What if you're in edit mode and some buttons appear, and you're not in edit mode and other buttons appear?"
>
> The consensus was in favor of that approach. I didn't like the approach, but rather than argue in a room with a bunch of people in front of a white board, I said, "Alright, let me mock it up, we can test some people, and if I'm wrong, fine."
>
> We mocked it up and tested people on a Macintosh, and sure enough people got confused. They said, "What happened to those buttons? They were there before."
>
> That's where the reductionist philosophy comes into play. We said, "Well, we can't fit seven buttons on here. Do we really, really need to have this button? Could you have it in menus?"
>
> "Well, now that I think about it . . ."

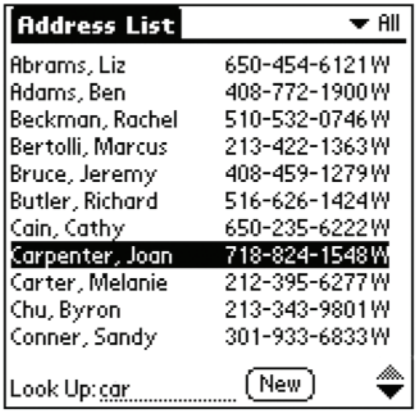
Stapler kept on the desk surface ■
Staple remover kept in a drawer ■

> That was how we started to think about how you use the product, and which functions you need all the time.

Rob Haitani came to the same conclusion about modes for the Palm OS as Larry Tesler had for the desktop,[23] but he did not always succeed in reducing the number of steps to a minimum for the first iteration. He wanted to optimize the address book, where he had discovered from the user studies that the most important feature is looking up a phone number:

> It's ironic that the address book was the one application in which we didn't achieve the one-step solution. If you want to look up a phone number in a PalmPilot, you have to pull out the stylus, and you have to know how to use Graffiti in order to do it fast and efficiently.

There is an inherent conflict between the desire to reduce the number of steps for fast access and wanting to minimize the number of items on the top level to reduce complexity. That drove Rob into a pragmatic philosophy of just trying to do the right thing on a case-by-case basis. The conflict was resolved because there are only a very few features in any application that people use very often. Take for example a word processor or a spreadsheet; there are only two or three commands that you use over and over again. The applications are therefore designed so that the very frequently used features are on the screen, and a menu, or another form of additional step accesses everything else.



Address List ▼ All

Abrams, Liz	650-454-6121 W
Adams, Ben	408-772-1900 W
Beckman, Rachel	510-532-0746 W
Bertolli, Marcus	213-422-1363 W
Bruce, Jeremy	408-459-1279 W
Butler, Richard	516-626-1424 W
Cain, Cathy	650-235-6222 W
Carpenter, Joan	718-824-1548 W
Carter, Melanie	212-395-6277 W
Chu, Byron	213-343-9801 W
Conner, Sandy	301-933-6833 W

Look Up: car (New)

Selecting a number from the address book

Rob articulates his design approach as a philosophy, but he also supports his philosophy with a very pragmatic design process: first understand the customer, then prioritize ruthlessly:

> One bit of advice that I gave to people designing the Palm OS was, "If you can really understand the one thing your customer wants to do most frequently, and make that a one-step process, then I guarantee people will like the product."
>
> Just say, "What is the one thing you want to do?" and even if you have to throw out conventions of logic, architecture, and hierarchy, you should make that one step. The more "illogical" your approach is, the less likely it is that it will blindly follow the conventional wisdom, and hence the more likely it becomes that you will be able to differentiate and create a successful product relative

> to your competition. If you take the conventional approach, by definition you're not innovating. If you just say, "Here are all the features," and you lay them out in a logical pattern, then that's not going to be a successful product.

The Palm Product Line

The PalmPilot was announced in February of 1996, complete with the new operating system, Graffiti, built-in synchronization, $299 price tag, fast processing, and the shirt-pocket-fitting form factor. Jeff Hawkins launched it at a conference, with one of the premier demo slots, but his time was limited to twelve minutes. He remembers how the projector went wrong:

> I got up there and I started describing the product, and it's clear that I had the audience's rapt attention. No one had ever seen anything like this before. It's small; people want something like this. Then I put it under the overhead projector, and I started the demo. The second that I started the demo, the screen went blank, but the clock was still running. You had your twelve minutes, and that was it. So I had six hundred people in the audience, and no one could see it. We tried for a few seconds to get the system to work, and I said, "I'm running out of time."
>
> So, I walked to the front of the stage, I held it up, and I said, "I'm going to describe to you what you would be seeing if you could actually see the demo."
>
> It was almost better than showing the real thing. It was like telling a story. I said, "You can see this small product, and when you push this button, it's going to instantly change to this screen, instantly change to that screen."
>
> When they got the projector working again we were out of time, but the audience clamored that they wanted the clock to be reset. So we had a chance to do the demo again. I put it back on the projector and I demonstrated what I had just told them, and it was great!

That first year the PalmPilot had very good reviews and sold about three hundred thousand units, which was excellent for an innovative product in a category that had no track record of success. It was also enough to get the attention of competitors, particularly Microsoft:

> Microsoft decided that they were going to kill us. They had conferences with a thousand developers in the room, where they put a big bull's-eye on the wall and in the center of the bull's-eye it said "Palm." Steve Ballmer stood up and said, "We are going to kill these guys. Don't even think about working for them."
>
> All the analysts started coming to us saying, "Well, you know, this is really too bad, I'm sorry you guys are going to lose."

Microsoft also applied their legal muscle over naming rights. The Pilot Pen Corporation had sued Palm successfully in Europe, where the trademark laws are more restrictive than in the USA, over the use of the name "Pilot," so Donna Dubinsky and her colleagues decided to abandon the name and replace "PalmPilot" with Palm followed by a number in roman numerals.

Initially Microsoft was calling their competitive product "Windows CE," but then they announced that they were going to change the name to "Palm PC." Jeff decided to fight back:

> We said, "You guys can't call it that!"
>
> They said, "When you said Palm, you meant the tree; you didn't mean a hand-held computer. Your original logo was a tree; you didn't even think about hand-held computing."
>
> I said, "You guys are crazy."
>
> They're the mega-company, right? What are we going to do? So, we sued them in Germany. They were launching the Palm PC in Germany at the Cebit show, and we got an injunction. The marshals came on to the Cebit floor and removed all of Microsoft's materials. It was great! So we got them to change; they had to change the name to Palm-Size PC. Eventually they changed again to Pocket PC.

By this time Jeff and his team had developed another product called the Palm III, but he felt that he needed something dramatic to respond to the threat from Microsoft; more products in the Palm line with incremental improvements would not be enough. The Microsoft approach was to throw all these software engineers at the problem, so their first version of the PDA software had many more features than the Palm OS, but Jeff thought that they did not add much in the way of genuine value for the user. He puzzled over the best way to respond:

> I said to myself, “What can we do that they can’t do?”
>
> The answer was, “We build the whole thing; they don’t build it. They just make software, and they get some other party to build the hardware. Let’s build a beautiful piece of hardware. They can’t tell someone to make a beautiful piece of hardware.”
>
> So, we decided that we were going to build the most beautiful, elegant handheld computer. We contracted with Dennis Boyle and his team at IDEO[24] to help do that, because that’s the perfect type of project for them. I think that was the first project that I worked with Dennis. I came in and I described what I wanted. I said, “This is going to be the defining product. This is what all future PDAs are going to look like. We want the most elegant and stylish product.”
>
> I actually forbade any new features in the software. The reason I did this was because when the product reviews came out, I didn’t want to have a comparison like, Palm adds three features and Microsoft adds twenty features. I wanted the only thing they could say be, “Beautiful!” And it worked.
>
> It was an iconic product and still is to this day—an icon for the PDA. The strategy worked beautifully; our market share soared. People fell in love with the idea of this as personal jewelry, personal style statement, and we just left Microsoft in the dust, because they were arguing, “Well, we have more features for this and this . . .” No one cared, because we had this beautiful iconic product, the Palm V.[25]

Next, Dennis Boyle tells his side of the story of the development of the Palm V, and the definition of the physical attributes that made it so successful.

Photo Author

Dennis Boyle

Dennis Boyle grew up in a family of seven children, four of whom ended up as engineers or designers. His mother was in science and chemistry, and his father was an architect who worked for an apprentice of Frank Lloyd Wright. He was always pointing out good design. In spite of the fact that the classes did not mix easily, Dennis studied both industrial design and mechanical engineering at Notre Dame in Indiana. He went on to the master's program in product design at Stanford, where he met David Kelley, and they stayed together from then onward, building a consulting firm in engineering product design that thrived on developing products for startup companies in California's Silicon Valley, and which later became IDEO. Dennis continues his connection to Stanford, lecturing in the program where he studied and learned to design and build products. He is famous for taking a new prototype of a design with him whenever he goes to a meeting. He has perfected the art of iterative prototyping as a way of accelerating the development process. He has always been curious about ingenious ways of making things, as well as clever or amusing designs. Over the years he has collected interesting examples of objects from wherever his travels have taken him. His collection became a magnet for the engineers and designers at IDEO, as inspiration for innovation could often be found by browsing among his objects. In 1990 the value of the collection was institutionalized as the "Tech Box," a combination of parts and materials library, database and Web site, which is duplicated in all the major offices of the company.

PalmV
3Com
26 Jan 99
S M T W T F S
8:00
9:00
10:00
11:00
12:00
3:00
4:00
5:00
6:00
7:00
8:00
New
Details
Go to
APPLICATIONS
CALCULATOR
abc
123

Dennis Boyle

Palm V

■ Palm V in recharging and synchronization cradle

Photo Steven Moeder

The talent that Dennis had for collecting things also showed itself in his habit of constantly acquiring the latest products on the market and trying them out. He was one of the first people to own an original PalmPilot,[26] and in spite of the fact that the first one had a lot of reliability problems, he became convinced that it was a great step forward because it combined synchronization with the PC, Graffiti, small size, and excellent interaction design for the operating system.

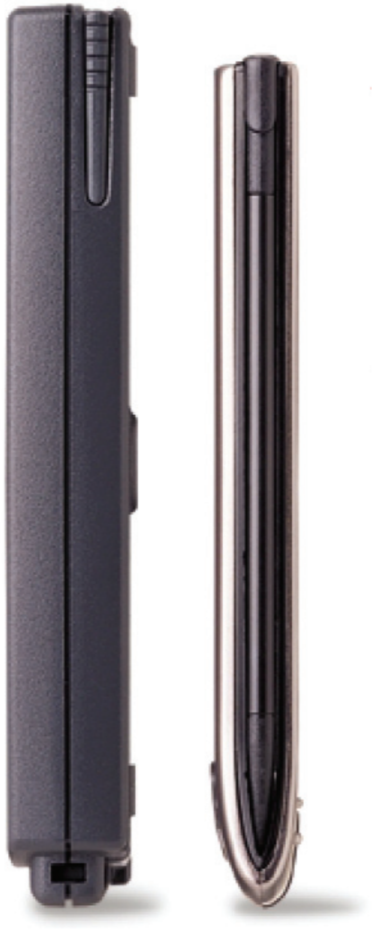
> I can clearly remember when I first became aware of the original Pilot in late April or May of '96. I had already tried a Newton, but it was just too big—it was not the right thing. Someone described it as a "high technology paving stone"; it was just too big.
>
> I said, "I'm not going to carry one of those electronic organizers unless it fits in my pocket and is more convenient." When I first saw the PalmPilot, I thought that it looked like something that I could really use.

Dennis patiently developed a relationship with Jeff Hawkins and Rob Haitani at Palm, building their confidence by showing them his many prototypes, and by his polite persistence. Jeff eventually decided to commission Dennis for the new version of the Palm series that became the Palm V. At the briefing meeting, Jeff flipped open a Motorola StarTac phone and was excited about the power of the design to captivate, saying that it just fit into your life because it was so elegantly thin and small. He pointed out that almost all of the sales of the organizers so far were to men, usually computer professionals in their thirties and forties, and asked, "How do we attract a bigger audience, and women in particular? What would be a much more natural fit for a wider group?"

The PalmPilot was nearly 20 millimeters thick. A goal of halving the thickness was high on the list, both to create a more elegant proportion that would appeal to a wider group of users and to make the fit to the pocket much more comfortable.

> We wanted something that didn't broadcast that you were using a piece of technology, or that you were overly concerned with technology. We were looking for something more like a piece of jewelry or accessory—an elegant watch or something you might wear, or have in your purse, like a compact or something. We put together a design team that was half women, which worked out very well.
>
> To get down to what turned out to be 11 millimeters (we almost made the 10 millimeter goal), every single tenth of a millimeter was fought over in a big battle. It became clear that plastic was just not going to work at all for the case. We made models that proved you could feel the difference; you could feel a lack of tortional rigidity, and if you dropped them they would shatter. Metal came out of the analysis and investigations as being a great choice.

It was hard to find examples of handheld products made out of metal to use as a precedent, but they eventually located the people who made the camera bodies for the Canon Elf, and worked with them to make the aluminum stampings for the Palm V.



Actual size thickness comparison between PalmPilot and Palm V

Dennis and his team noticed that people were making accessories for the original PalmPilot, and sticking them on with tape and Velcro, as there was no provision for attaching things in the design. This led to the idea for two identical rails, on either side, so that accessories could be attached. On one side you could keep your stylus, depending on whether you were right- or left-handed, and on the other you could attach a protective case or some other accessory.

The timing for the Palm V was perfect, as there was a fast-spreading desire for an electronic organizer that was inexpensive and conveniently small, and the Palm OS was gaining a reputation for ease of use, with the reassurance and convenience of synchronizing with a desktop computer. At that time there was very little competition, in contrast to the flood of competitive products that arrived in the following years. Over five million Palm Vs were sold, making it the highest-selling single model of PDAs.

Despite this success, Jeff Hawkins felt compelled to leave Palm and found Handspring.

Handspring

Back when Jeff Hawkins was inspired to start from scratch with the development of the PalmPilot, he needed much more in the way of resources and investment than if he had stayed with the original business plan of just writing software for PDAs. Donna Dubinsky, in her role as president of the company, was looking around for a major investor to provide the funds to develop and launch the Pilot and ended up selling Palm to US Robotics. The relationship worked out well, until US Robotics was in turn acquired by 3Com, which paid a high price for the modem business, without paying much attention to the fact that Palm was part of the purchase. A year and a half later the modem business was almost worthless, but Palm was worth quite a bit.

3Com was a networking company and had nothing to do with handheld computers, but they were happy that the success of Palm was making up for the losses of the rest of the company.

Palm V—front view
Handspring Visors, with Springboard expansion slot

visor
CityTime
San Francisco 11:55 am
New York
Th 2:55 pm
London
Th 7:55 pm
Tokyo
Fr 4:55 am
Paris
Th 8:55 pm

They needed profits from Palm but were unwilling to invest because of financial pressures on the modem side, just at the time when the Palm business was growing fast and needed resources to expand to the next level. Jeff remembers the increasing tension caused by the lack of investment:

> We said, "Look, eventually Palm has to be separate from 3Com. It can be now, it can be three years from now, but we're in a very different business."
>
> We decided to leave when the management of 3Com made it clear that they would not spin out Palm. About ten people from the original team who had started Palm, including Rob Haitani, left with Donna Dubinsky to continue on what we'd started at Palm. The easy thing for us to do at Handspring would have been just to compete with our old company. We could have done better products, we felt we could out-market and out-design them, but we didn't want to do that. We did not want to kill our old friends, so we said, "What are we going to do that's different?"

Jeff had been convinced for years that the future of handheld computers would be in wireless communications, but the right technology to use was not yet obvious:

Handspring
Visor Edge

Photos
Steven Moeder

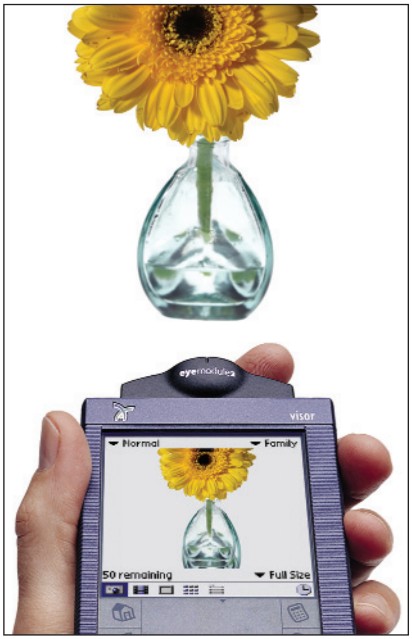
> At the time we started Handspring, it was very clear to me that eventually all handheld computers would have wireless communications in them. It was not clear to me at all that cell phones would be dominant; I didn't even know about cell phones at the time—it wasn't even in my vocabulary—but I knew that some form of wireless was going to be important. I wanted to try to figure out how to incorporate wireless into handheld devices, without committing to a particular technology.
>
> I felt that if Handspring built a wireless device immediately, we'd likely fail, so I wanted to build a mainstream organizer, but somehow add a strategy that would allow wireless. We decided to put in a slot in the device where you could plug in different radios. I actually built another wooden model at the beginning of Handspring, and it had a slot and all these different radios I could plug into it to see what it would be like. We thought it might be two-way pager, a Wi-Fi equivalent, or a cell-phone radio. Then we said, "Well, if we're going to do a slot, let's make it a generic slot, so people can put anything they want into it."

> We created this expansion slot. It was a success and a failure at the same time. It was a success because we attracted a lot of people to create add-ons for the Handspring, with what we called the Springboard expansion slot. It was a failure because even though we sold a lot of PDAs with the slot, we didn't sell a lot of expansion products. We sold a lot of memory cards, but most of the other expansion things did not sell well, but we did use it as originally intended for learning about different radios.

eyemodule

Jeff was still relying on Dennis Boyle and the IDEO team to develop physical designs for the new organizers at Handspring, all of them including the Springboard expansion slot. First was the Visor, then a color product called the Prism, and later a thinner product called the Visor Edge, similar in concept to the Palm V.

Dennis was aware of the expansion concept from early on, and he decided to take advantage of the fact that he was helping to design the expansion slot and to develop the first example of a plug-in module in parallel; the plug-in was a camera called the "eyemodule." To stay true to Palm's philosophy of simple interactions, he decided to create a camera that you could plug into the slot, and it would immediately convert the PDA into a camera. He describes the interaction design solution:



Eyemodule being used in Handspring Visor to capture photo of flower in vase

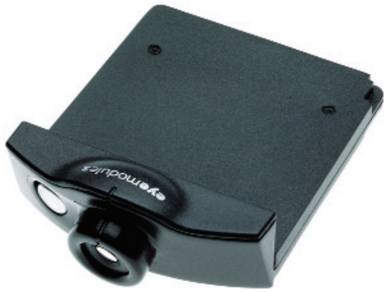
> It was an attempt to have our own little business and take advantage of the platform. It was clear that cameras were going to be part of the future for handheld products, but in 1999 the technology for miniature cameras was immature. I am quite proud of the fact that you can take this module, and plug it in, and you have no action to do. When inserted, it starts up automatically and goes into the camera mode within five seconds. All you have to do if you want to take a picture immediately is to press the button, and a picture will come up.
>
> In order to create this simple-seeming result, there were several layers of complexity in the design solution, so I put together a multidisciplinary team. First we had to research and develop the

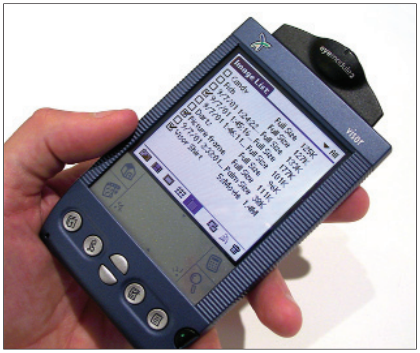
technology for the camera that would meet our price point and fit into the geometry of the slot. Next our human factors psychologists did some user studies about the way people thought about taking pictures on a PDA, and came up with a framework for the interactions with three stages—capture, view, and share. The interaction designers created icons for these three modes, so that they popped up on the bottom of the screen of the organizer. They were designed to fit into the visual and behavioral style of Rob Haitani's operating system and to use as little space as possible, leaving most of the room for the image that the camera was looking at.

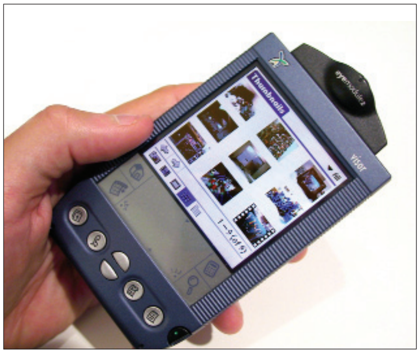
For the "capture" component of the design, we created an interface to reside on the PC to allow you to sort and edit images, and to use the communications of the system to send copies to other people. We also put up a Web site to sell the "eyemodule" and to support customers, and we developed the packaging and user manual as well. Then there was the physical design of course, and the manufacturing, so we involved just about all the disciplines at IDEO in the process.

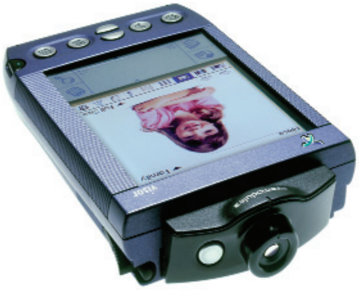
Jeff Hawkins was pleased that we developed the "eyemodule" in parallel with the Visor itself, as it turned into a very useful device to demonstrate the versatility of the expansion slot concept.

The next project for Handspring was the Treo, combining the PDA organizer with an email communicator and a cell phone.









Eyemodule removed from Visor ■
Eyemodule in Visor—photo list ■
Eyemodule in Visor—thumbnails ■
Eyemodule in Visor—front view ■

trēo
handspring
powered by
News
Stock Quotes
Sports
Yellow
Direc
Weath
Directory of Mobile
Google S
Copyright © 2001 Handspr
trēo
VICE
drew Meyer
31-555-6737
OnHold
00:53
Cooper
Active
00:05
Hang Up

Combining PDA and Cell Phone

During the development of the Palm platform, the original research that Rob Haitani had conducted with focus groups had made Jeff Hawkins decide to postpone the integration of telephone and email functions into a PDA. Back then, although the technology had evolved to the point where it was technically possible, the trade-offs needed to make it happen were prohibitive. As Rob said, "It would be the worst phone and the worst organizer, combining the worst of all worlds."

Five years later, the technology was maturing to the point where the team at Handspring thought that the time was right to combine a PDA and phone, and that they could use the integration to advantage rather than having to make problematic trade-offs, so the Handspring Treo was conceived. Jeff describes the impetus for convergence:

> We were starting to realize that the cell phone was going to be the king in the world, and that it would become the design center for people's personal communication devices. We wanted to bring the benefits of the organizer, all that stuff we'd done there, to the cell phone. Up until then, there was no cell phone with a platform or third-party software.

■ Handspring Treo

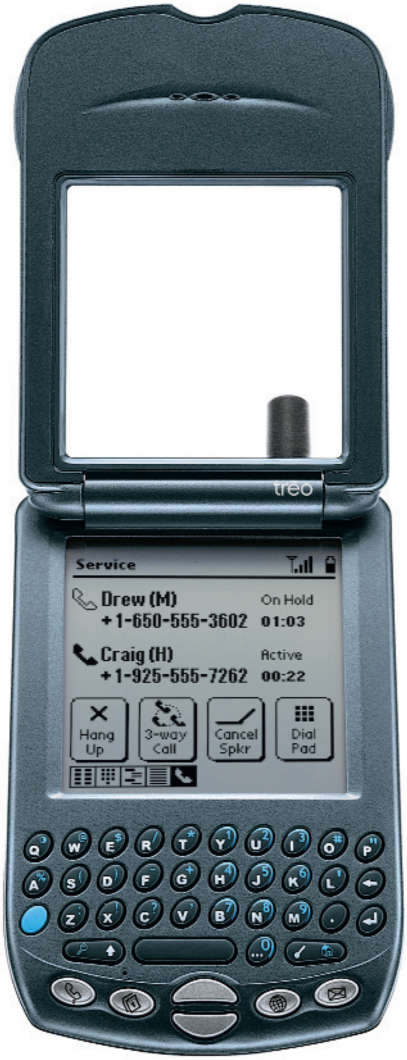
Photo Courtesy of Handspring

The first Palm OS products were essentially organizers connected to PCs, and a cellular phone is a completely different animal. It was necessary to reinvent the user interface, using the same design principles. If you want to call someone, you just want to grab the phone and make the call, so Rob Haitani used his rules of ruthless optimization to look up phone numbers. He developed algorithms so that you could simply open the lid and type three or four letters, find one name out of a thousand, and be immediately connected.

One of the problems of integrating an organizer with a phone is that you want to look at your organizer while you are on the phone. If you are talking to someone, how do you schedule a meeting and continue to talk? That problem was solved by integrating a speakerphone, so that if you're on the phone you just press the spacebar to switch to speakerphone mode, allowing you to maintain a conversation and switch to any application in the organizer without having to put the phone to your head.

Another very simple design feature was a little switch on the top that let you turn your ringer off without taking the Treo out of your pocket or clip, thereby shortening the moments of embarrassment when you have forgotten to turn off your ringer in a public place.

When you are using a Treo for wireless email, you want to reply to messages as if you were using a full computer, demanding much more fluent text entry than for normal organizer tasks. Dennis Boyle remembers the surprising success of the miniature QWERTY keyboard:



■ Treo with QWERTY keyboard

> When you compare Graffiti to entering text with buttons in some fashion, it shows how market needs change over time. There was almost no good way to enter text in a small handheld device. From the very beginning, with Apple, when they tried to do handwriting recognition, it turned out to be so poor that it became kind of a joke among writers and cartoonists.
>
> It was clear that people needed some sort of text input. Hawkins had the idea of meeting it halfway by developing Graffiti, a stylized alphabet that people could get pretty good at with a couple of hours of practice. It captured some devotees, and it was enough to get the business going, but it was still a very small percentage of the population who could even be bothered doing that.
>
> I remember that Handspring decided to put the Treo out with both a keyboard and Graffiti, because they didn't know which one people would choose; they decided to let them vote. The result was quite clear; a large majority went for the small keyboard. It doesn't seem intuitive, when you show it, people say, "How can you possibly type on that?"
>
> Then I just type out a little paragraph for them like this in a matter of seconds, and they say, "Wow, you get pretty good at that, don't you?"
>
> The human hand and eye can be trained quite quickly, and it seems like the little QWERTY keyboard, bad as it is, is such a standard that it requires no guesswork, and that attracts more users. The

> success of the RIM BlackBerry products, followed by the keyboard version of the Treo, proved that some sort of little thumb keyboard could be effective. It broke through a "thumb barrier," if you will.

The keyboard was a very dramatic change in interaction style. Rob Haitani had always known that Graffiti was something you could learn and pick up, but also felt that at some point they would hit a wall in the number of people who would be willing to learn a different way of writing. There are people who just prefer keyboards. They want buttons; they want to press an *A* and get an *A* rather than deal with a stylus. The challenge posed by the keyboard was that it immediately projects an image of a much more complex product. How do you overcome that and keep the same essence of simplicity? Perhaps people are ready for more complexity.

Jeff Hawkins has an amazing track record of clear forward vision, but even he is uncertain enough to want to rely on iterative development, rather than expecting to be right first time:

> In designing products, it's really hard to know what the best design center is. I always try to imagine the future, but usually that takes a couple of iterations. It took us a couple of iterations with the PDA and it took us a couple of iterations with the Treo.
>
> The first design for the Treo was a clamshell. It had the keyboard, it had the screen, it had Palm OS, and so on. It was a reasonably successful product. We sold 350,000 of them and it won a lot of acclaim, but it wasn't a huge success. We didn't sell enough, and it wasn't gaining enough traction. But we took that knowledge and said, "Okay, let's do another tweak. Let's try to hone in further. What have we learned?"
>
> The next try ended up being the Treo 600, which I think is much closer, and I believe it will have a long life. Its design center will have a very long life; I have not seen a replacement for it yet.

Treo 600 with built-in camera ■

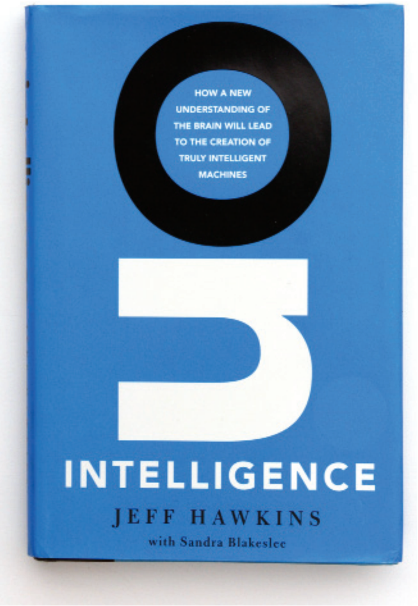
Jeff is sure that the cell phone is the dominant communicating device of the future, and he uses statistics to emphasize the point:

> The cell phone has already won. There are six and half billion people on this planet. Somewhere between 1.2 and 1.5 billion of them have

> a cell phone. That is the number of people actually using cell phones today [in 2004]. The PC doesn't come close. Nothing else comes even close. It's done! Game over. A few years from now, I don't know what it will settle out to—maybe 3 billion? But it's incredible. Think about the impact on humanity. It's an amazing thing.

Throughout his career Jeff Hawkins has flip-flopped between designing computers and studying the brain, and in 2004, after the launch of his new book *On Intelligence*, he reflects on the connection between neuroscience and design:

> Brains like familiarity, but they get bored. They are genetically programmed to want to discover new patterns. You don't want it too new because that seems dangerous. You want it somewhat familiar and somewhat new.
>
> Think of music. The best music has some kind of essence of things you can recognize: a normal beat, harmonies, and melodic phrases, but you don't want to hear the same old, same old. You want something that's slightly jarring, and a little bit clever. The newness matters more than any other particular aspect of the aesthetic value. You want newness combined with cleverness.
>
> Somehow new and old at the same time gives the best design. If a design is so new that people can't relate to it, then they reject it, even if they could theoretically learn how to use it because it's very clever. Styles are like this in general; if you have a new style for clothing, generally you don't want it to be too crazy. You want it to be just slightly different, enough that people say, "Oh, that's cool."
>
> It's built into the human brain. We want familiarity, we want to be able to learn how to use it, but we also want some newness to it, and that's what makes us excited about it.


HOW A NEW UNDERSTANDING OF THE BRAIN WILL LEAD TO THE CREATION OF TRULY INTELLIGENT MACHINES
INTELLIGENCE
JEFF HAWKINS
with Sandra Blakeslee

■ *On Intelligence* by Jeff Hawkins

How surprising that Jeff achieved his first great success with the PalmPilot by insisting on design simplicity in a complex world, but in 2004 he is looking for his next great success by combining PDA with cell phone, email communicator, and browser—not a simple combination:

> I want to continually try to move toward the best single product I can build. It doesn't mean that doing multiple things is bad. People thought that the PalmPilot was successful because we only did a few

> things; that's not true. The things we did, we did well. People don't mind doing more things; they like it as long as you do them well. As long as you make it easy, intuitive, fast, and so on, they don't have a problem with that. That's the distinction. I think we can do a lot. And I think it's inevitable that the phone in your pocket will be most of the world's Internet connection, most of the world's communications, most of the world's email, most of the world's Web browser. It's going to be their music player, it's going to be their video player, eventually, in a number of years, it will be their PC!

In the next chapter, "Adopting Technology," we look into the reasons behind these paradoxes of familiarity and boredom, simplicity and sophistication. David Liddle discusses three phases of the adoption of technology, and the interviews that follow show some examples.