Background Paper on Pen-Based Computing

By Timothy Scanlan, Inventor of Ring-Writer

2001

*In the early 1990's, pen-based mobile computers were seen by many as the "next generation of computing," a "paradigm shift," which would radically transform the way people used computers for work, and for fun.

A new Operating System, PenPoint, was designed and implemented by GO Corporation, and Microsoft responded with Pen Windows. Some early machines were built using one or the other system, and sometimes both, and many small start-ups blossomed with the promise of a totally new consumer market.

In the United Kingdom, Active Book Company built a machine based on an ARM chip, and with a Smalltalk O/S, only to be later absorbed by At&T into the short-lived EO Corp, along with GO. EO disappeared, with some 50 to 100 million dollars of venture capital, leaving Apple to take up the banner of pen-based computing, and to define a new class of machine, with the Newton PDA.

Newton failed to cross the mass-market chasm, and since then pen-based computers have effectively been restricted to niche markets. What happened? And is it possible that things are sufficiently different now, to imagine re-inventing this market, like Nintendo re-invented video-games after the market founded by Atari, Mattel Electronics, and others, effectively self-destructed?

In this paper, we shall attempt to define what a real Mobile Computer should look like, what it should do, what enabling technologies are available now, and what needs to be built in order to make the original vision a reality.

What is wrong with Portable PC's?

We first need to consider what is already available for people on the move. Perhaps "Mobile Computing" was just some large-scale vapourware of the early nineties. Let's start with weight, shape and size.

Mainstream Portable PC's are still too heavy: it is true that some "ultralights" have appeared on the market, but the typical Portable PC is still more a "Transportable PC". The machines are designed to go from flat surface (office or home desk) to flat surface (meeting-room table, hotel-room table, and so on). They are too heavy (and hot) to be used on one's knees, and in any case the typing position would be too awkward to consider any prolonged use in a "relaxed" position. Use in aeroplanes is so inconvenient as to restrict serious use to the reviewing of documents, and has often more to do with status-display than with working efficiency. And who has ever seen someone using a PC standing up?

Shape and size are determined by the keyboard. Only Apple's iBook departs convincingly from the standard rectangle, and even in this case, the size of the machine is clearly determined by the need to accommodate keyboard, touch-pad and selector-

button. The width of Portable PC's is of course practically invariable. Smaller machines do exist, but anything between a standard Portable PC and a handheld like the various Psion models has been of marginal market interest, and were really just handhelds masquerading as real computers.

So, we can perhaps already conclude that today's Portable PC's are not really "mobile" machines. How do they fare then as transportable desktops? Well, not very well, especially for use over long periods of time. The keyboards are usually not detachable from the rest of the unit, and are not angled to provide a comfortable typing position. The keys are accessible only over the touch-pad / selector-bar combination, and can hence be awkward to use efficiently. Modern screens are generally good, but usually inferior to what most people use as a desktop monitor.

On a completely different note, the upright screen can become a "barrier" when the device is used in a meeting or a consultation session. This is particularly unfortunate for people in the medical profession, or in sales situations.

A final word concerning the touch-pad / selector bar: it would be difficult to invent a credible interface for pointing and selecting that is worse than this abomination! In terms of machine real-estate, the interface's footprint is, well, about the size of a foot, taking something like half of the available surface area. Surely, we can do better than this!

The truly amazing fact is that these answers to our original question are probably not very different to the answers that we would have come up with ten or even fifteen years ago! Portable PC's are suited neither for desktop use, nor for portable use: they look like a fairly poor compromise between a "desktop" workstation and a "mobile" computer.

What was wrong with the first generation Mobile Computers?

The original vision of the pen-based pioneers was mostly correct: a book-sized machine, with a powerful pen interface, aimed at mobile professionals. Indeed, the vision gained many adepts at the time, including several large corporations (IBM and AT&T amongst others), and venture capital firms. Clearly, nobody thought that the ultimate pen-based machine would be built in the first generation, but as it turned out, the early machines were so obviously far from the ultimate vision, that confidence in the market completely evaporated.

PenPoint was a good Operating System. The book metaphor was elegantly implemented, and the basis was laid for third-party developers to build useful pen-based applications. It is true that Pen Windows was in contrast a fairly ugly, ill-adapted extension to Microsoft Windows, which did nothing to take any real advantage of the pen interface. This failed adaptation no doubt did much to discredit pen-based computing at the time.

It is true that the early pen-based machines were hopelessly underpowered (RCA even courageously released a model based on an 80286 microprocessor!). However, adequate machine power would have become available within a matter of a few years, and this problem wasn't in itself sufficiently serious to totally invalidate the technology. Two major failings condemned the first generation pen machines to the role of impressive, but mostly unusable, technological gadgets.

First, the new systems totally underestimated the importance of providing an adequate means for entering text. The machines were intended for general-purpose use, not just as limited “electronic slates” aimed exclusively at form-filling applications (which is one of the main niches that pen-based computers have occupied since). Someone wanting to use a pen-based computer as a sort of mobile desktop would expect to do quite a bit of text entry. Unfortunately, the pen systems designers effectively abandoned this critical aspect of systems usability to third parties, in the pious hope that someone else would come up with the miracle recognizer that they were unable to implement. Unfortunately, no one else knew how to solve the problem either!

To make matters worse, having disowned the text input interface, the pen systems designers failed to provide the system structure that would have allowed others to implement novel word-based pen interfaces. As far as text input was concerned, the systems treated the pen as a keyboard simulator. To a certain extent, this was perhaps also a result of putting too much emphasis on the eternal bugbear of “backwards compatibility”, and not realizing that the pen interface was truly a paradigm shift that justified a new approach to applications interface design.

The second major failing was nobody’s fault. A great part of the work of any mobile professional at the time, as well as nowadays, involved the use of electronic mail. Pen-based machines had the same limited communications facilities as Portable PC’s: the user had to wait to be next to a telephone plug (and hope that he or she had the correct physical connector) in order to send and receive e-mails. Mobile communications were too far away in the early 90’s (they are still a couple of years away now) to have given any market advantage to the first mobile machines.

Two contrasting cases: Apple’s Newton and the Palm Pilot

During the last decade, following the initial resounding failure of what we might call general-purpose pen-based PC’s, we have been able to observe two very different attempts to use a pen interface in a consumer device. It is worthwhile to compare some key aspects of these two pen machines, in order to deduce some guidelines for future products.

Let us try to understand why the Newton was not a success, despite the formidable advantage of the Apple trademark. In this respect, we can note already that some 200,000 Newton’s were sold, and that many of the owners were still very enthusiastic about the machine at the time that Apple decided to discontinue it. This market performance would have been a remarkable success had the Newton been commercialized by a start-up, since the Newton incorporated several groundbreaking and risky technologies. However, in the Apple context, the experiment was seen as a failure. What, with hindsight, can we say about the reasons?

First, the marketing message wasn’t clear. “Personal Digital Assistant” sounded very good, but no one really knew what that meant. There was lots of AI hype about “Newton Intelligence,” but from the applications point of view, the machine really just looked like a miniaturized PC, or rather a handheld computer, similar to many already on the market. The big difference was of course the pen interface, and the cursive recognizer incorporated into the system.

The Russian company Paragraph provided the recognizer used in the Newton. It was

based on research undertaken in the late 70's, and was probably the best cursive recognizer in the world at the time. This fact did not prevent technologically ignorant journalists from lampooning the technology, without taking account of the fact that this was a very competent attempt to solve a very difficult problem, and the first time that cursive writing was proposed as an input method.

The real problem, no doubt, was that the recognizer had no chance of living up to Apple's pre-release hype. Added to this, Apple had done nothing to prepare users for this new input method. For a first test of the recognizer, most people chose to write their name. Now, a cursive recognizer, or any word-based input method, is effectively incapable of recognizing words not in the dictionary, and unless the person was called something like "Bill Gates," the recognizer would propose something totally ridiculous in place of the unknown words. In most cases, the user went away with a very negative first impression of "Newton Intelligence."

Even for users aware of the problems cited in the previous paragraph, cursive writing recognition cannot provide a viable alternative to the keyboard. Let's spend some time understanding the reasons for this.

In order to achieve a recognition rate of better than 90% on a cursive recognition system, the writer must take particular care in the formation of his or her words. This still leaves something like one word in ten to be corrected. The need to write especially carefully, plus the time taken correcting the erroneous words, results in an effective input rate that is unacceptable for most users. Indeed, the input rate is probably less than the same user could achieve using a much less sophisticated single-character recognizer.

Should the user try to write faster, the recognition rate will fall, more corrections will be needed, and the effective text input rate will be even lower. So the fairly limited speed advantage of cursive writing (about 25% over hand-printing) is totally eliminated by the inadequacies of the system, even for the relatively rare users who are capable of adjusting their writing in order to accommodate the recognizer!

Indeed, only a small minority of users (perhaps a quarter, or a third) are even capable of achieving high recognition rates with cursive recognizers available today (which are only marginally better than those available in the mid-90's). The vast majority of users discover that the machine cannot recognize their normal handwriting even when they make some effort, and they often give up disgusted, disappointed, or both.

One of the other problems confronting the use of cursive recognition for text input is that handwriting is an unconscious activity, and as such is difficult to modify. Most people, confronted with a cursive recognition system for the first time, have no idea what kind of adjustment they need to make in order to be recognized. Compare how you would write a letter to a small child, with the appearance of written notes that you might take at a conference. For many people, the personal notes would be illegible (unrecognizable) if presented out of context, or at a time too far removed from their creation. Humans do not recognize handwriting, they <u>read</u> it, and no computer system has yet even come close to doing this.

In contrast to the marketing hype accompanying the launch of the Newton, the Palm Pilot made apparently modest claims. It offered relatively few standard applications, but those that it did offer worked elegantly and well, and served a definite user need (in

particular, the electronic diary and the address book). In addition, the connection to the desktop for file transfer and data synchronization was very convenient.

The machine had an attractive design and was of appropriate size, fitting comfortably into a shirt pocket (the fact that the early models were bought overwhelmingly by or for men may have had something to do with this!)

Finally, the text input system worked sufficiently well for the needs of the principal applications. Let's look into this question in a little more detail:

It is very difficult to implement even a single-character handwriting recognition system that will handle not only the full range of lower-case and upper-case letters, numbers, punctuation signs and other symbols, and which will also take account of the many different ways that people form all these characters.

Graffiti, the character recogniizer used on the Palm Pilot, was invented to a certain extent to avoid the harder recognition problems. Letters and numbers are handled separately, and upper-case is achieved by using a "shift" gesture. However, the key innovation of Graffiti was to decide not to try to recognize natural handwriting, but rather to use an artificial alphabet built from a mixture of lower-case letters, upper-case, and some invented symbols (usually having some resemblance to the letter they replaced). In this way, the worst letter-pair confusions were avoided, and it was fairly easy to implement a recognizer that would efficiently handle this reduced character set.

But the real advantage of Graffiti was much more subtle. The typical reaction to handwriting recognizers up to Graffiti was "it didn't recognize my writing!" Graffiti turned the tables on users, so that when a letter was not recognized, they were practically forced to admit that they hadn't correctly formed the symbol, and apparently, they didn't mind! Added to this, since using Graffiti wasn't quite writing, there was less interference between Graffiti and the user's natural writing style, and so less need to "unlearn" automatic hand movements.

So, what is a Mobile Computer?

Let us now imagine what a new generation of mobile computers might be like, taking into account the lessons learned from the failure of the first generation, and the possibilities opened up by new technologies.

Many of the ideas of the first generation pioneers are still good. The pen is the ideal interface for a mobile device, and the book metaphor is probably the most natural document interface. In any case, mobile machines will certainly be "book-sized" - that is almost any size that a book can be! Product designers, freed from the constraint of the keyboard, will be able to propose machines with shapes and sizes appropriate to their intended purpose, from communicator wristwatches to electronic canvases for artists.

The Mobile Computer should not attempt to be a general purpose mobile desktop. It should look like a consumer product with a rugged functional exterior, and should be usable in just about any environment. It must be easy to understand and simple to use. It must be easy to install: unpack, install batteries, and switch on. The applications should stay active between shutdown and re-activate: no tedious system loads. The user needs a simple way of protecting his/her data, perhaps by means of signature

verification, and the system should provide good network security. While compatibility with desktop systems is neither required nor desirable, the mobile systems must provide easy document transfer and synchronization with the desktop.

A few years ago, people thought of a PC as something used for preparing reports and spreadsheets. That has changed significantly: nowadays, most people would say that the prime functions of computers are getting information (the Internet) and sharing information (electronic mail). Mobile communications must be the key focus of mobile computers.

Within the next few years, high-bandwidth mobile communications will become possible: this key enabling technology was sorely lacking for the first generation machines. Mobile computers have the potential to be everyone's main window on the Network. It is foolish to imagine that users will be happy to surf the net on a telephone screen the size of a postage stamp: yet that is what the phone companies and operators are telling us. The new communications possibilities would be much better put to use if the user can deal with page-sized pieces of information.

So perhaps the "killer applications" of Mobile Computers are quite simply the Internet and electronic mail. Certainly, the Operating System should be designed to make those applications work exceptionally well, just as the Palm pilot was designed for the Calendar and Address Book. Add to these a word processor that really takes advantage of the power of the pen, both for input and text correction, and incorporate a multi-modal user-interface that will allow other novel input methods to be available for the user. All applications must then be not only "pen-aware," but also ready, when appropriate, to cater for these different input methods and their associated raw data.

A completely new range of services could be provided to the user, who will be considered as almost always connected: news services, specialist information services, targeted but user-controlled advertising, and many more. Imagine down-loading your favourite newspapers and magazines before a plane journey, or loading a selection of electronic books for a vacation trip.

The user should not be forced to think in terms of files and directories. Documents should be easily retrievable by using keyword search, as well as by name. It should be impossible for a user to lose more than just the latest modifications to a document, all archiving and back-up being provided as a network service. Software updates should be available by simple request, or should arrive automatically as background traffic. With Mobile Computers, we should at last be truly able to say "The Network is the System".

Requirements of the input system

The preceding sections have shown that, far from being technological gadgets, pen-based book computers could now indeed fulfill the dream of a new generation of mobile devices, opening up a new world of communications possibilities to all manner of users, both technical and non-technical.

Of the two missing enabling technologies that were identified, the necessary wireless communications infrastructure will soon be in place, but there is still no good replacement for the keyboard. Some of the hopeful hand waving that assumed fifteen years ago that handwriting recognition would do the job is now, to some extent, being

applied to voice recognition. It is highly unlikely that voice recognition will solve the input problem by itself for some time to come. Let's see why.

Voice recognition systems have the same disadvantage as cursive handwriting recognition: even a collaborative and practiced user has difficulty in achieving recognition rates approaching 90%. Allowing for some technological improvement, we might hope for this figure to be attained by more people, but the software has already reached a stage of greatly diminishing returns.

Even with this optimistic figure, however, the correction problem is, if anything, even worse than for cursive recognition. Repeating an unrecognised word is more than likely to give the same wrong result, and spelling out the words letter by letter just removes the problem to a new even more frustrating level, where "m" sounds like "n", and "s" sounds like "f". The answer might be to associate voice recognition with a good single-character recogniser, but the effective text input rate will still be slow, even for those who know how to talk to a voice recognition system.

Voice recognition systems will not work well in a noisy environment, and there are many situations in which voice recognition could not be used, like conferences, meetings, or any environment where discretion and tact are required. But perhaps worse than all this is the fact that people just do not like talking to machines, as witnessed by the millions of uncomfortable voice messages left on voicemail systems every day.

Given that none of the "obvious" solutions will solve the problem, something new is required. The paper that follows describes Linear-X, a writing system designed specifically for communication with a computer. The system answers the two critical requirements of a text entry system: that the system be easy to learn by the novice, while providing the potential of rapid text entry for the user who is prepared to invest time in order to become proficient.

The system is not trivially easy to learn, like handwriting systems were imagined to be. On the other hand, the time required to become proficient would be much less than the time required to learn to type adequately, or the time that everyone invested in order to learn to write. The reward for learning Linear-X will not just be the pleasure of acquiring a new technological skill, but instead the basic ability for using a next-generation computer, a mobile window on the Network.