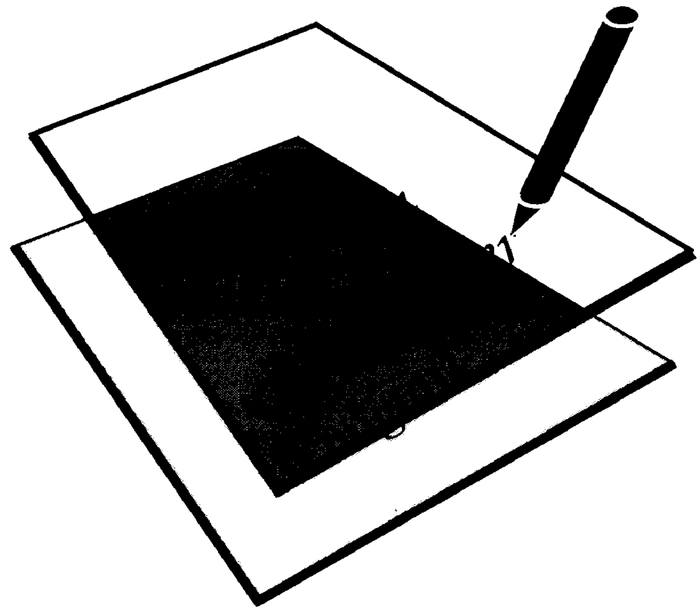
Special Feature



Pen Computing

A Technology Overview and a Vision

André Meyer

Beneath the rule of men entirely great
The pen is mightier than the sword.
Baron Edward George Bulwer-Lytton, *Richelieu*, 1838

This work gives an overview of a new technology that is attracting growing interest in public as well as in the computer industry itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of interaction between a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions.

Starting with a short historical background and the technical advances that begin making Pen Computing a reality, the new paradigms created by Pen Computing will be explained and discussed. Handwriting recognition, mobility and global information access are other central topics. This is followed by a categorization and an overview of current and future systems using pens as their primary user interface component.

Table of Contents

Keywords

pen-based, stylus, displays, digitizers, low-power processors, PCMCIA, handwriting recognition, unistrokes, Jot, mobile computing, ubiquitous computing, wireless communications, Object-Oriented Dynamic Languages (OODL), Object-Oriented Prototype-based Languages (OOPL), tools, assistants, agents, PenPoint, Newton, General Magic

1 Introduction

Pen Computing lives both in our best-known real world environment and in a world of vision.

The idea is simple. Use your primary and best developed human skill – writing by hand on a sheet of paper. Man would not be able to exist without language and writing. Sharing written and drawn information is the most important means of cultural exchange. And today we're even better at exchanging information – we use computers. But computers do not understand the most simple strokes of our beloved gestures. Computers are actual illiterates when it comes to recognizing hand-drawn writing.

During the time of evolving computer technology many different means for user input have been invented: keyboards, mice, trackballs, lightpens, etc., but none of these is really fitted the human way of communicating thoughts and ideas.

This is where the Vision – referred to in the subtitle of this work – comes in: computers that are able to sense our drawings and interpret our handwriting would be a lot easier to use and much more useful than the currently known input devices. The ease of information manipulation would result in an intensified concentration on contents, with the user being freed from technical restrictions.

In practical terms, a pen-based computer consists only of a flat display which records and displays the traces from a user's moving pen. The pen replaces a keyboard and other input devices. Were it not for the additional output capabilities, e.g. video and sound, bystanders would not necessarily recognize this as a machine but as a paper notebook. The computer is also able to recognize handwritten text, i.e. it can convert scribbles to printed text. Pen-based computers exist in various sizes and for a multitude of application classes.

The following sections will provide more detail on the necessary technical components and on the consequences in respect to human-machine interaction paradigms.

- Section 2 *History*
  This section starts with some fundamental thoughts about the role of language and writing and outlines the developments of languages and writing tools. Briefly, a history of Pen Computing is presented.
- Section 3 *Hardware*
  In this section, all the components that are necessary to build a pen-based computer are presented and discussed: displays, digitizers, processors, batteries, and expansion cards. The latter two are especially important for mobile pen-based systems.
- Section 4 *Handwriting Recognition*
  Handwriting recognition is discussed mainly from the user interface perspective. The algorithms used for recognizing handwritten scribbles as text are complex. The most important methods are introduced very briefly. A different approach that is aimed at high recognition rates at the cost of using a simplified character set (unistrokes) is illustrated, as well as a standard for storing hand-drawn input (Jot).
- Section 5 *Mobility*
  Mobile systems are an important application area for pen-based systems because they can be used almost anywhere. This is in contrast to keyboard-based systems. This section gives an overview of various wireless communications systems that are currently used or being developed. The difficulties that arise from accessing data that are highly distributed as a result of mobility are also discussed.
- Section 6 *Paradigms*
  Pen Computing introduces new paradigms for human-machine interaction. First, the paradigms for designing the user interface are presented, followed by a round-up of programming paradigms. Finally, a categorization of hardware and software architectures is outlined.
- Section 7 *Current Systems and Products*
  This section presents the most interesting pen-based systems that are currently available. Compared to the other sections, they can be taken as a measure for the state of the art in pen-based technology.
- Section 8 *The Worldwide Real Virtuality*
  All things put together, this section builds the climax of this work by reasoning about the consequences that can be followed from integrating the information of the previous sections. The Vision referred to in the subtitle of this work is enlightened here.

2 History

2.1 Writing

2.1.1 Written Information

Information needs to be written down to make it permanent, while the permanence of information is a requirement for making it communicable. The entire set of means for communicating information in a particular context is referred to as culture. The process of assigning a meaning to a signal (a signified to a signifying) makes up for the specifics of a certain culture, together with the selection of the signifying representations of information, i.e. what kind of visual expressions are accepted and understood. Culture is possible only when individuals communicate their ideas and visions by writing them down for other people and for other times and places, pointing out the super-personal nature of culture. "The laws of communications are the laws of culture" [Eco, p. 38].

Of all the means of cultural exchange, writing is the most specific and precise, as well as the most flexible. A pictorial representation leaves much more room for interpretation. Theodor W. Adorno explains the relation like this: "What crackles in works of art is the sound of the friction of the antagonistic moments the work of art seeks to bring together; writing not at last because, like in the signs of a language, its processuality is encrypted in its objectivation."[1]. They are based on the same process, but the process works at different levels of consciousness. Writing is a notation based on a language and therefore well defined. On the other hand, deciphering written information is more complicated and demanding.

Common to all kind of communication is the division in syntax and semantics and with this the establishment of semiotic codes [Eco]. [Lévi-Strauss 1, pp. 31–41] describes this relation on a more general level as the interaction between structures and events[2].

Table 1 ([Eco, p. 86]) gives an interesting illustration for the variability of semiotic codes in semantic systems.

Table 1: Variability of semiotic codes for the semantic field *wood–tree–forest*

<table>
<tr><th>French</th><th>German</th><th>Danish</th><th>Italian</th></tr>
<tr><td>arbre</td><td>Baum</td><td rowspan="2">trae</td><td>albero</td></tr>
<tr><td rowspan="2">bois</td><td>Holz</td><td>legno</td></tr>
<tr><td rowspan="2">Wald</td><td rowspan="2">skov</td><td>bosco</td></tr>
<tr><td>forêt</td><td>foresta</td></tr>
</table>

Written information is useful and important only as long as it is understood correctly. As semantic systems vary over time and place, it is often difficult to understand the messages. This is also the reason for the difficulties encountered when texts are translated between languages. Therefore, languages are a very powerful as well as a very delicate means of expression.

2.1.2 History of Writing Tools and Styles

The history of the culture of writing and writing tools starts with the need for languages to be written down and communicated. This is possible only with languages that use individual letters and characters instead of pictorial representations for entire words or expressions[3], essentially beginning with the Semitic writings (historically known in the form of early Sinaic since about 1800 B.C.E. and Phoenician since about 800 B.C.E.). The current Hebrew alphabet and writing is known to have been popularly used since about 600 B.C.E. Later – about 400 B.C.E. – Greek was developed, mostly as a variation of the older Phoenician). Greek was the first script that was written from left to right. From Greek followed the Byzantine and the Roman (later Latin) writings. Latin also has roots in the Etruscan writing (about 300 B.C.E., written from right to left). In the beginning, Latin had only uppercase letters like all of the other writing systems mentioned so far. Minuscules were developed only far later (around 600 C.E.) when the writing instruments were refined enough for such detailed faces (cf. figure 2).

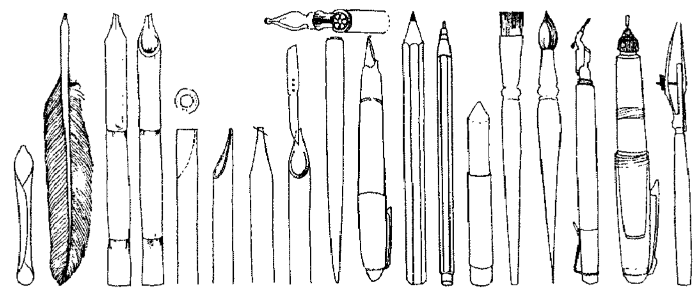


Figure 2: Writing tools

Many variations were developed during the centuries, most of them based on late Latin variations, e.g. Gothic and Humanist writings.

While in the beginning the letters were hammered in stone or pressed into wet clay with wedges and the appearance was rough, later cut wooden sticks and then bird's feathers[4] were carefully prepared to write on papyrus and on parchment with ink. The use of feathers and brushes allowed for a more detailed writing face. Around 1800 metal "feathers" were used[5]. Later again, pencils[6], pens[7], ball-point pens[8], and others were invented.

The substrate on which to write was also subject to many changes. Stones and clay were replaced by papyrus and later by parchment (around 1300 B.C.E.). Paper was invented in China in 105 C.E. but it became known only much later, due to Chinese secrecy – in Japan around 700 C.E. and brought to Spain by the Arabs in 711. But paper was widely used throughout Europe only after paper mills were built in the late 14th century.

Paper became the primary medium after another dramatic change took place: the invention of printing with replaceable wooden or metal letters by Johannes Gutenberg in 1436. Simpler kinds of printing, e.g. stamps with names, were used much

1. "Was an den Kunstwerken knistert, ist der Laut der Reibung der antagonistischen Momente, die das Kunstwerk zusammenzubringen trachtet; Schrift nicht zuletzt deswegen, weil, wie in den Zeichen der Sprache, ihr Prozessuales in ihrer Objektivation sich verschlüsselt." [Adorno, p. 264]
"Die Fixierung durch Schrift oder Noten ist der Sache nicht äusserlich; durch sie verselbständigt sich das Kunstwerk gegenüber seiner Genese: daher der Vorrang der Texte vor ihrer Wiedergabe." [Adorno, p. 153]
2. Lévi-Strauss compares scientific reasoning (generating structures – hypotheses, etc. – to test them with the events) to mythical reasoning (overlaying structures to observed events), both valued equally and with the arts as a combination of the two (simplified) [Lévi-Strauss, p. 36]
3. Some writings, like Chinese and Japanese, have obvious roots in pictorial systems, but they were developed further to a level where they could overcome simple picture sequences. They are, therefore, not excluded from this definition.
4. Geese feathers were used from about 500 B.C.E. to our days.
5. Metal "feathers" were already used in Ancient Rome, but only in 1808 B. Donkin in England applied a little cut where the ink flows on the substrate, which made them more comfortable and precise tools.
6. Independently invented in 1790 by Nicolas-Jacques Conté in France and Joseph Hardtmuth in Austria.
7. The first real pen with a container for ink was invented by L. E. Waterman in New York in 1884.
8. This invention came from the Biro brothers in Hungary in 1938.

earlier in China, but they did not find their way to Europe. During the centuries, many newer printing technologies were developed based on Gutenberg's printing machine, e.g. offset printing.

On the other hand, when printing could be automated, the process of text input later was also revolutionized, first by Henry Mill in England in 1714 with the invention of a typewriter, followed by many variations and improvements, e.g. daisy-wheel typewriters by IBM in 1961. Another major step forward was the separation of text input and printing. This could be achieved when the texts or graphics could be stored by use of computer memories. The text and graphics could then be entered by use of a keyboard or a mouse and printed later at any place with the right printing facilities[9]. Now, it is possible to eliminate these bulky input devices again and to use the same method as men used for centuries to write texts and draw pictures: write with a pen on paper – with the only difference that the paper is electronic paper.

2.1.3 Writing and Literature

"The pen – this rapidly moving little servant of human intellect"[10] is of course a very good friend of those who do not write for simple information exchange – taking notes, remembering meetings, etc. – but who write for its own glorious sake: the poets and writers, whose pens pave the paths to imagination. Some of them also write of their relation to the tool they require so much, although nowadays only a few writers still write by hand. The pen may be understood metaphorically in this case.

The poet's eye, in a fine frenzy rolling,
Doth glance from heaven to earth, from earth to heaven,
And as imagination bodies forth
The forms of things unknown, the poet's pen
Turns them to shapes, and gives to airy nothing
A local habitation and a name.

Shakespeare, *A Midsummer Night's Dream*, 1594

In cases where a writer attempts to express his inner self in words there may occur a certain, almost pathetic, relation which is, though, never harmful nor dangerous.

No man was more foolish when he had not a pen in his hand, or more wise when he had.

Samuel Johnson, *Of Goldsmith*

2.2 History of Pen Computing

The first computer scientist who wrote down the vision of a computer that could be used effortlessly by untrained users was Alan Kay [Davidson], [Ryan 3]. In 1968 he envisioned the Dynabook, of which he fabricated a cardboard model. The Dynabook was a light-weight device on which one could take notes and work interactively with wireless communications. In 1972 he helped build up the Xerox Palo Alto Research Center (PARC) where he wanted to make the Dynabook happen. Influenced by developments in the early sixties[11] the team around Alan Kay, Adele Goldberg, Larry Tesler and others developed object-oriented programming (OO), the graphical user interface (GUI), and personal computing (PC) – Smalltalk was born. But it was not until the Mac that these inventions were widely spread. Kay is an Apple Fellow (i.e. a free sponsored researcher) since 1984.

A prototype similar to the Dynabook was presented by Apple in 1987: the Knowledge Navigator, which included the book design of the Dynabook and added speech recognition and intelligent information retrieval (agents). An heir of the Knowledge Navigator, but a real-world machine this time, was shown in 1992 by ex-Apple boss John Sculley: Newton. It had started in 1987 as a project of Steve Sackoman and is now a major part of Apple's personal interactive electronics department (PIE).

There were many other people and places in the history of Pen Computing among which the most remarkable are Robert Carr and GO Corp., who developed PenPoint as the first truly pen-oriented operating system in 1990, Nicholas Negroponte's MIT Media Lab, who invented Multimedia, and Mark Weiser's research group on ubiquitous computing at Xerox PARC.

All of these visionaries share their concentration on contents and users instead of technological feasibility.

3 Hardware

In order to realistically think about Pen Computing it is necessary to consider the underlying technical problems that have to be solved for building a computer with the following capabilities: the intended machine should start at the lowest common denominator of writing – a block of paper and a pencil. A computer with the same capabilities must be able to

- track the position of a pen
- display the ink left on the writing surface
- overlay texts and drawings seamlessly.

In addition to this it should

- recognize the writing
- provide endless paper
- store the drawn pages and search them.

The following sections cover the state-of-the-art in the technology of displays, digitizers, processors, batteries, and extension cards.

9. Postscript should be mentioned here because it allows for a portable description of a page's contents.

10. S. R. Hirsch in his commentary to Vayeira.

11. 1956 RAND tablet and first handwriting recognition
1957 Dimond's *Stylator* and first on-line handwriting recognition
1960 mouse by Douglas Engelbart at Stanford University
1963 interactive graphics (*Sketchpad*) by Ivan Sutherland at the MIT [Sutherland]

3.1 Displays

3.1.1 Requirements

Writing is only possible on a flat, paper-like surface. This points out the need for very high integration of all the parts of a pen-driven machine. The ideal is to build a writing surface that contains a computer, and not vice versa.

First, one should consider the requirements for an idealized display that suits the needs of Pen Computing:

- The size may vary from a few centimetres for hand-held devices to the size of a double newspaper page (60 cm diagonal) or a wall board (3 m diagonal).
- Flatness and weight are the most obvious requirements for portability and for large-size displays. Displays as thin as paper should be envisioned.
- The screen update must be fast for the integration of animated graphics and video.
- The resolution (density of picture elements, or *pixels*) should be equal to that of a high-quality printer (at least 240 dots/cm).

3.1.2 Implementations

Currently, two different display technologies exist.

- Emissive displays with light-sources generate light in order to display pixels.
- Light-valve displays block or transmit either ambient light or backlight.

The first category is best represented by the well-known CRTs (cathode-ray tubes). They excel in many areas: brightness, contrast, color representation, resolution, reliability. They allow a wide viewing angle and they are relatively cheap, and widely used. On the other hand, they require a lot of space, are heavy, consume a lot of power, and they produce x-rays and magnetic fields. All of these negative aspects make CRTs unacceptable for use in Pen Computing. Flat-panel displays (FPDs) [Baran 1, 5], [Crystaloid], [Hallett], [Hayes], [Werner], [Woodward, Long] rely on the second, much younger technology. They are light in weight, very thin, and consume only little energy. All of these attributes make them very well suited to make the ideal come true.

Flat-panel displays consist of two glass plates with active display elements located in-between. On the inner sides of both plates stripes of transparent, conductive electrodes are laid out orthogonally for addressing a matrix of picture elements where the electrodes intersect[12].

- Passive-matrix displays apply voltage row by row and for each row the pixels are set to the brightness according to the column voltage applied. In this manner, all the rows are activated sequentially for the whole picture frame. Although the refresh rate is usually high, there is a little flickering effect because the pixels are activated for a shorter time than it takes them to be refreshed. The effect gets bigger with increasing screen sizes.
- Active-matrix displays use switching devices for each pixel, which are switched using the row voltage while the intensity is set by the voltage applied at the column axis. Every pixel stores its voltage in a capacitance until it is refreshed or changed. This way, flickering is almost eliminated. The functionality is very close to that of DRAMs[13].

A VGA size display with 640 x 480 pixels needs as many horizontal and vertical electrode stripes; for color displays triple the amount of stripes is needed, totalling up to 921600 picture elements. None of the pixels must be damaged as the eye is sensitive to the slightest irregularities. This high standard in quality is difficult to achieve.

3.1.2.1 Twisted Nematic Field Effect Liquid Crystal Displays (TNLCD)

Liquid crystals are used in liquid crystal displays (LCDs) – an oily substance containing molecules (Cyanobiphenyl) that orient themselves along the lines of electrical fields. The most common type of LCDs are TNLCDs (cf. figure 3), with many newer variations discussed later.

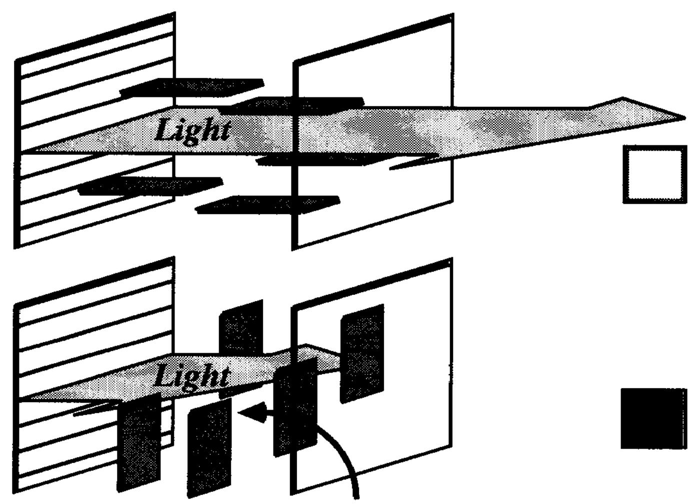


Figure 3: Schematic rendering of a liquid crystal display, observer positioned at right side.

TNLCDs use molecules that twist horizontally from 0 degrees without voltage to 90 degrees when voltage is applied – thus, they transmit or block light to create a picture pixel by pixel according to the electrical field that is applied. This happens when the incoming light which passes a polarizing filter, is either aligned with the rotation of the molecules at 0 degrees and thus can be reflected. In the case when the polarized light is oriented orthogonally to the molecules (at 90°) no light can be reflected and the corresponding pixel appears dark. The same procedure applies for all angles of the molecules between 0° and 90° as well, resulting in various levels of grey.

12. Indium-tin oxide (ITO) may be used for the electrodes as it is transparent and conducts electrical current well.

13. Dynamic random access memory (DRAM) is a fast memory chip technology.

Colors are produced with three different color filters (usually red, green, and blue) that are laid over the three corresponding picture elements that form a single pixel. The loss in light intensity that stems from these additional filters is compensated by backlighting the display, i.e. there is a light source located behind the glass plates to overcome the habitual weakness of ambient light.

3.1.2.2 Passive-Matrix TNLCDs (PMLCD)

PMLCDs use molecules that react slowly[14] to the applied voltage in order to reduce flickering. On the other hand, animated images appear very blurred. Their contrast is poor and the viewing angle is limited.

3.1.2.3 Supertwist Nematic LCDs (STNLCD)

STNLCD's molecules twist more than those of ordinary TNLCDs to ameliorate contrast, viewing angle, and resolution. For double supertwist nematic LCDs two layers of liquid crystals are put on top of each other to further improve contrast and viewing angle. They are therefore heavier and more expensive.

3.1.2.4 Active-Matrix LCDs (AMLCD)

AMLCDs feature pixels that are constantly activated to avoid flickering and improve image brightness. AMLCDs are usually fabricated with thin-film transistors (TFT) which are fast switching, allowing for animated images to be displayed without blurring. In fact, these TFT-displays are large ICs with millions of switches laid out on glass instead of silicon. The larger size is responsible for the higher costs, because the yield in perfect displays is still low.

3.1.2.5 Ferro-electric LCDs (FLCD)

FLCD's crystals are bistable, i.e. ferroelectric liquid crystals hold the applied electric field until it is changed. In this way, they behave like the crystals in TFTs, but without the complex switching mechanism, making them easier to produce, and they are even faster[15].

The difficulty with producing FLCDs is that the two glass plates must be only 1 or 2 microns apart (previously mentioned technologies work well with gaps of 5 to 7 microns) and it is difficult to display grey scales due to their hysteretic behaviour.

3.1.2.6 Plasma-Addressed LCDs (PALCD)

PALCDs are similar to active-matrix TFTs but instead of the complex TFT layout and switching, channels containing gas are used for each pixel row. The gas is ionized by electrodes and a plasma reaction with the column electrodes activates the individual pixels. Capacitances hold the activation for the time of the frame refreshment, which is very short. Thus high frame rates are possible even for large displays.

3.1.2.7 AC-Cycle Plasma Display Panels (ACPDP)

ACPDPs produce light by ionizing gas that achieves the plasma state and then glows in orange color. Illuminated pixels remain in the plasma state until they are extinguished, resulting in bright and flicker-free images without complex reactivation, suitable for even very large displays[16]. Other advantages are the high speed[17], high contrast, and wide viewing angle.

The difficulty with producing grey scales and colors is still a major problem for plasma displays. A different gas must be used for each of the three primary colors. It looks as if these problems could be overcome.

3.1.2.8 AC-Cycle Thin-Film Emission of Light Displays (ACTFEL)

ACTFELs produce light by electrically exciting phosphors. These give images similar to those of CRTs in quality. The phosphors for full color displays are not yet available, but the rest of the production process is similar to common IC production (layering techniques are used), only larger and with different materials.

3.1.2.9 Field-Emission Displays (FED)

FEDs are similar to CRTs, with the difference that they are flat and use not only one electron gun for the whole display, but one gun per pixel. The traditional, though flat, phosphorized screen with a mask is combined with a flat panel of cathodes with a gap of only 0.2 mm in-between. The cathode panel consists of a matrix of very fine needles which emit electrons as a result of the field effect. One pixel consists of three picture elements for the primary colors, each of which is addressed by electrons emitted by around a thousand needles. The intensity of the field effect is so great that the FEDs consume much less power to obtain the same brightness as a conventional CRT.

Hence, there is no flickering, although the image quality of a CRT display is maintained: very bright, high-resolution, low-power displays even with less x-ray and magnetic field emission. This technology is comparatively old and has not made a breakthrough yet, but since it was recently re-engineered at LETI[18], Grenoble, France, it seems to be gaining more attention.

3.1.3 Conclusions

Flat-panel displays are not only needed for Mobile Computing. Larger screens, e.g. for High-Definition Television (HDTV), become a problem with CRT technology, because the space requirements for CRT devices are about equal to their screen size times its diagonal in depth.

Currently, TFT displays are the leading technology for commercial flat-panel displays. Active-matrix LCDs offer the best and fastest images. Xerox, for example, demonstrated an AMLCD with 113 pixels/cm, i.e. the same as a standard laser printer (300 dpi), with a total of 1536 x 1024 pixels, even in color. Unfortunately, AMLCDs are still very expensive. Passive-matrix LCDs are much cheaper and there is hope for improving image quality by a new technology: active-addressed passive-matrix LCDs address rows of the image by calculating pixel values in respect to the other pixels, thus avoiding crosstalk (smeary stripes). The pixel values are stored in a sep-

14. 100 to 200 milliseconds
15. 100 nanoseconds
16. Currently up to 150 centimeters.
17. Currently up to 60 frames per second.
18. Research laboratory of the French atomic energy institute.

arate memory chip and not in the display itself as is done with AMLCDs. Nevertheless, LCDs are sensitive to temperature, shock, and vibration, which makes them a problem for ubiquitous use. They are also not reliable enough yet, in production, as well as in use.

Plasma and phosphor displays, on the other hand, are reliable, long-lived, and rugged and they do not need backlighting, but the image quality is not yet perfect, especially for color displays. Plasma displays are, however, the most promising alternative for large displays and color quality will improve significantly over the next two or three years.

The very interesting FEDs are still not commercially available, but several companies plan to introduce them in the years to come.

In the future, CRTs will begin to vanish with the advent of flat-panel displays that are large, cheap, rugged, and offer a good image quality. Other display technologies that are coming en vogue, e.g. projection systems, are less suitable for combination with digitizers and for mobile use.

3.2 Digitizers

3.2.1 Requirements

The most obvious requirement for a pen-based machine is that it captures hand-drawn input. A digitizer does just that.

Many different digitizer technologies [De Bruyne], [Ward, Schultz], [Ward, Phillips] have been developed since 1956, when the first digitizer was implemented by Rand Corp. In the beginning, digitizers were opaque tablets lying on a table and the input was represented on a CRT screen. These tablets were quite large and a digitizing puck was used as the principal means of input. Pucks are suitable for digitizing single coordinate points, used in computer-aided design (CAD) – the major application of digitizers. Pens were used only rarely, because free-hand input was seldom required. Pens are less precise than pucks, because the angle between the tablet and the pen varies in time and between different users.

With the development of Pen Computing, pens became the major input device and the digitizing tablet had to be combined with a flat-panel display. Hand-drawn input is now possible and a must for applications such as handwriting recognition. The combination of digitizer and display allows for a much higher interactivity – like writing on paper.

Of course, this improved interactivity does not come free. There are two ways in which a digitizer and a display can be combined. If the display is mounted on top of the digitizer, the digitizer's accuracy diminishes due to the increased distance from the digitizing grid to the pen's tip. On the other hand, if the display is fitted under the digitizer, the display's contrast is less and the digitizer must be transparent.

Only some digitizer and display technologies work together, limiting the choice of existing respective technologies dramatically[19].

Speed[20] and precision are needed for digitizers that are used for Pen Computing, especially when it comes to handwriting recognition (HWX). The requirements of an input device for different user actions may be summed up as in table 2.

Table 2: Requirements for digitizers (inspired by [Foley, Wallace, Chan])

Action / Requirement	fast	precise	Examples
selection			icons, menus
data point entry		•	CAD
data path entry	•	•	HWX, signatures, drawings

3.2.2 Implementations

3.2.2.1 Electromagnetic Digitizers

A grid of conducting wires covers the active area of an electromagnetic digitizer[21]. This grid generates a magnetic field because the wires are looped. Inside the pen a wired coil is hidden that induces a current flowing through the grid's wires next to it by means of a magnetic field it generates. The coordinates of the grid are determined by the strength of the current and then interpolated for higher accuracy. Accuracy is also a function of the density of the grid's wires. Another method for improving accuracy is to lay out two wire grids with different spacing and interpolating between the two.

Instead of the pen generating a magnetic field on its own (active pen) another implementation uses a passive pen with a coil that needs not be powered and thus allows for the design of lighter, cordless pens. The tablet then must send and receive signals to and from the pen, changing its magnetic field at very short intervals[22] and it therefore consumes more power than a design with a tethered or battery-powered pen.

Among the errors that electromagnetic digitizers frequently produce are the following: When the pen moves diagonally at a fast speed, a velocity error occurs as a result of the delta time between the sampling of two respective x-/y-coordinates. Instead of a straight line a curve appears in one of the directions. A problem with the distance of the coil from the grid arises when the pen is tilted. The same line is interpreted differently when drawn with the pen held at different tilting angles. The electromagnetic field can be disturbed by external magnetic fields due to interference. In the corners of the tablet

19. Color displays have not yet been combined with digitizers successfully.
20. The stream of x/y-positions sent by the digitizer's controller must exceed 120 points/second for good handwriting recognition.
21. The Rand tablet was already built with this technology.
22. 20 μs each for sending a radio signal and receiving the pen's induced signal.

the magnetic field typically is distorted and therefore less accurate than in the centre.

A flat-panel display must be in front of the digitizer because the wire grid darkens the image too much.

3.2.2.2 Electrostatic Digitizers

Very similar to electromagnetic digitizers, electrostatic digitizers differ in that the grid's wires are not looped but connected on only one side of the tablet and that the pen's coil is replaced by a capacitive probe. No magnetic field is therefore generated.

When the pen's probe (a metal tip) approaches the grid, which generates electrostatic signals, it receives a signal which it transmits through a tether to the tablet controller. The controller calculates the coordinates from the current position of the wire that sends the signal.

This technology, with the pen used as the receiver, eliminates the tilt error, but it is more affected by electromagnetic disturbances. Therefore, the display must fit under the digitizer to shield it from the computer's electromagnetic fields.

3.2.2.3 Resistive Film Digitizers

Resistive film digitizers consist of a dielectric, insulating material, e.g. glass, coated with a transparent, slightly resistive conductor, e.g. indium tin oxide. On one edge of the conductive coating a voltage of 5 V is applied and on the opposite edge a voltage of 0 V. In between, the voltage range is linearly decreasing from 5 V to about 1 V. When the pen touches the tablet, its metal probe takes up a certain voltage which determines its position in one direction. The same procedure is done for the second direction in order to obtain a coordinate pair for the pen's location.

Another design uses a pen that emits a 5 V signal that is caught up by receivers in the four corners of the tablet film. The respective currents determine the coordinates in this case.

A different approach, similar to these designs, determines the coordinates with two film layers, through which current flows in orthogonal directions. The same declination in voltage results from the films' resistance. The two film sheets are separated by a dielectric liquid or similar substrate. The coordinates are determined by the amount of voltage that can be read at the two film sheets' edges when they are pressed to touch each other. A passive pen can be used with this purely pressure-sensitive design, or even a finger nail.

Resistive film designs are easily added on top of existing displays, and they are comparatively cheap. The film, however, reduces image clarity resulting in a lower contrast, and is not well protected against scratches. Some designs add a protective sheet on top to protect the film glass. Another problem is that, unlike the previously discussed designs, resistive film tablets cannot sense pen proximity. The pen either touches the tablet or is not detectable. Some systems profit from the information of pen proximity, e.g. to start with handwriting recognition when the user lifts the pen off the tablet.

3.2.2.4 Capacitive/Electrostatic Film Digitizers

Again, an indium tin oxide film is added to a transparent substrate, glass or plastic, but in this case the film lies under the substrate to prevent scratching it. The film may be very thin and thus is transparent enough not to decrease the contrast of the underlying display. The electrostatic field is applied at the opposite edges of the film sheet and the pen receives the signal from the surface of the glass. It reads the electrostatic field as a whole, as opposed to the pen used with resistive films, where small defects in the film render regions of the digitizer useless. At the same time this also reintroduces the capability for sensing proximity and the technology is very power efficient.

3.3 Processors

3.3.1 Requirements

Processors for pen-driven devices are subject to some specific restrictions [Pountain 2], [Ryan 4], [Statt], especially when they are supposed to be portable:

- Low cost.
  The pen-oriented user interfaces are suited for people who do not want to spend a lot of money for computers or who do not want to buy computers at all. Only low cost devices have a chance for success in the mass-consumer market. A price of about US$ 25 per chip set would be appropriate for building mass-consumer-oriented devices, assuming a price below US$ 500 for the whole unit.
- Smallness.
  If the chip set around which a hand-held device is built takes up too much space, there is no chance any person can actually take it along in a pocket. Earlier, it was already mentioned that the 'computer' should only be part of the writing surface and not be visible by itself.
  The size of the silicon chip influences the printed-circuit-board space and the die size of the chip determines its price by the amount of silicon used.
- Low power requirements.
  There is a general trend to move from 5 V towards 3.3 V architectures [Prince], [Shepard] as these low voltage designs have several advantages: lower voltage allows higher clock speeds (less heat is generated: heat ~ I ~ U^2), lower energy consumption (by about a factor of 5), and smaller and lighter silicon chips.
  State-of-the-art 64-bit, 100 MHz architectures are only feasible using voltages lower than 5 V because of the generated heat and the required space. The standard for 3.3 V (±10% tolerance) was publicized by Jedec (U.S. Joint Electron Device Engineering Council) already in 1984 and has gained general acceptance in the industry.
  By merely letting processors run at a lower voltage, there are no benefits at all – apart from saving energy. Running 5 V chips on 3.3 V reduces the clock speed and therefore the computing power. But architectures designed for lower voltages can use smaller transistors and run at higher clock speeds. The transition time from logic '0' to '1' is shorter and the channel length can be reduced from 0.5 μm to about 0.4 μm for higher integration.
  The power consumption for desktop CPUs is about 3 to 5 W, for low-power CPUs it lies in the range of about 100

to 300 mW. Power can be used even more efficiently by built-in power management that lowers frequency and voltage when the system is less active. Architectures using CMOS technology are fully static, i.e. the clock speed can be reduced without losing the processor state. Some architectures provide instructions to stop the CPU and continue only timing for peripherals, others have a stand-by mode that stops everything to reduce power consumption down to a few microwatts.
In mobile devices where batteries are used as the energy source, power management is crucial. Battery life extends 3 to 4 times when operating at 3.3 V instead of 5 V. On the other hand, there is a 40% saving in weight for the same battery life using only 3 instead of 5 battery cells (cf. section 3.5 for more details on batteries).
For architectures that require even less power (1–2.7 V) no standard has been proposed yet, but such architectures exist in many variations. On the high performance side, there are plans for a DEC Alpha chip running at 2.5 V with a frequency of 300 to 500 MHz. The only problem with low voltage is that peripherals such as hard disks and displays may possibly not run at lower than 5 V.

- Sufficient computing power.
  The feedback from a pen-driven machine must be immediate. No delay in ink representation is allowed. If handwriting recognition is desired, the real power must show up, because this is a very computation intensive task. With the use of several independent recognizers (e.g. for text and graphics) that work simultaneously, there is a need for parallel execution that demands high computational performance. Handwriting recognition should even work in the background without bothering the user at work and it must be fast for immediate response. The required computing power can be achieved by using fully 32-bit RISC[23] architectures with support for clever memory and power management.
- High integration.
  High integration means that much functionality that is otherwise split up into a whole set of chips, e.g. peripheral drivers and controllers, should be put together on one single chip. This makes the design of a new device much easier and the extensive supported standard functionality allows for better customized solutions. Higher integration also satisfies demands for more processing power and more memory.
  The development in integration is impressive: Intel 80386 chips were built using about 375000 transistors, the 80486 chips used about 1.2 million transistors, and with state-of-the-art low power architectures up to 40 million transistors per chip are possible, allowing for the integration of Multimedia functionality, voice recognition, handwriting recognition, etc. on-chip.
- Compact code.
  Compact code saves both ROM and RAM memory by using shorter and more integrated instructions. There are several different concepts for using compact code: the use of 16-bit instructions in spite of or in conjunction with 32-bit instructions, variable-length instructions, complex instructions (e.g. test and branch, test and add, etc.) supporting the pipelining features of RISC processors.
- (Non-) Compatibility.
  Compatibility is not required for most pen-driven systems. The mobile pen-driven devices mostly do not replace desktop machines, but with their supreme ways of communicating, data can be exchanged without a need for explicit application or system software compatibility. Compatibility is of concern for exchanging data only. This enables designers to use leading-edge technology and to support more user-oriented features.

23. Reduced instruction set computers (RISC) are more powerful than the older complex instruction set computer (CISC) architectures, because they dispense with the complex instruction sets of CISCs – the comfortable complex instructions are seldom used but require a very complex architecture – and implement only the simple instructions that occur often and are fast. Unified clock cycles – ideally one instruction per cycle (except for jump and branch instructions or load and store operations from and to registers and memory) – allow the CPU to execute several parts of an instruction (instruction fetch, instruction decode, operand fetch, instruction execute, store result) in parallel (pipelining). A large number of register are responsible for the renunciation of direct memory operations (load/store architecture). A lot of the difficult work is made by the compiler instead of the hardware (e.g. register allocation, instruction ordering, etc.), i.e. at compile time rather than at execution time.

3.3.2 Implementations

3.3.2.1 Intel 80x86/Pentium

These processors are used for compatibility reasons. They do not meet any of the above mentioned criteria, but there are architectures such as the AMD Elan Am386SC that incorporate many of the mentioned features while remaining compatible. The Elan is said to run a mobile-oriented version of Microsoft Windows. There are, however, a wide variety of pen-based systems running Microsoft's Windows for Pen Computing on Intel processors (cf. section 7.1).

3.3.2.2 AT&T Hobbit 9201x/9202x

The Hobbit chip set [AT&T] consists of 5 chips: the ATT92010 is the central processor with a built-in memory management unit and on-chip stack, the ATT92011 (System Management Controller) controls the whole chip set including power management, interrupt control, address decoding and access to the local bus structure, the ATT92012 (PCMCIA Controller) controls up to 4 PCMCIA cards, the ATT92013 (Peripheral Controller) builds the interface between the Hobbit local bus and up to eight 8- or 16-bit peripheral devices compatible with the ISA bus standard, and the ATT92014 (Display Controller) supports CRT and LCD displays with resolutions of up to 1024 x 768 pixels with 1, 2, or 4 bitplanes and knows four different display modes to save power (normal: screen and video RAM are updated regularly; static: screen is updated, but no data is written into video RAM; screen off: screen is switched off, but memory is retained; display off: screen and memory are switched off). The chips run at 3.3 V and yield 13.5 MIPS running at 20 MHz, using 515 mW. CRISP (C-machine Rational Instruction Set Processor) 32-bit technology is used, a mixture of RISC-like performance with CISC-like code density. Together with the local bus structure this chip set provides for the functionality of a whole mobile device. The Hobbit chip set is a complete solution caring about power management and with enough processing power, but future versions will be more integrated. The newer Hobbit 9202x chips go that way: the 92020S is a faster version of the 92010, the 92020M is a three-chip set, and the 92020MX/92021MX are a two-chip set providing the functionality of the original five-chip set and a higher performance.

AT&T are currently evaluating combinations of the Hobbit product line with wireless communication elements for building improved mobile communications devices. The AT&T Hobbit is used in EO's Personal Communicators (cf. section 7.2).

3.3.2.3 ARM610/710

Advanced RISC Machines, Ltd. (Cambridge, U.K.) [van Someren, Atack] [ARM], has designed the ARM6 architecture as a macro cell. The CPU kernel fits on the corner of a silicon chip and allows the rest of the silicon to be used for implementing other functionality. The ARM610 is a full-featured 32-bit RISC processor with 31 registers and a three-stage pipeline. It integrates a hardware multiplier and measures 22 x 22 x 1.4 mm, has 144 pins, and gives a performance of 15 MIPS. A sophisticated memory management unit is integrated on the ARM610 chip. Advanced memory management is required for controlling memory access and garbage collection – a necessity for object-oriented operating systems. The code for the ARM610 is very compact because many common operations are built in and can be accessed as instruction options (e.g. add-and-shift and conditional branches are one-cycle instructions). Therefore, it excellently supports high level languages. The fully static implementation is responsible for the low power consumption.

The newer ARM710 processor has a larger cache (8 kB), yields 25 MIPS, and uses 33 mW running at 33 MHz. The ARM 610 builds the core of Apple's Newton MessagePads (cf. section 7.3).

ARM's market position has improved recently as a result of an agreement with Digital Equipment (DEC). The two companies intend to develop a more powerful line of low-power/high performance RISC processors known as StrongARM. DEC has gained considerable experience in powerful RISC design with its Alpha processors.

3.3.2.4 Hitachi SH7034

The Hitachi CPU integrates 4 KByte of RAM, 64 KByte of PROM, a DRAM controller, DMA engine, serial ports, an eight-channel A/D converter, and several timers all together on a single chip. The CPU also features a 16 by 16 bit integer multiply and accumulate instruction suitable for fast digital signal processing. A second version, the SH7032, differs from it in the amount of on-chip memory: 8 KByte of RAM and no PROM.

This chip alone could build up a mobile device, in conjunction with a display controller and a PCMCIA controller.

3.3.2.5 NEC V805/810/820

The V805 is a 16-bit version of the 32-bit V810. Both integrate a single precision FPU on-chip. The V820 additionally features DMA, serial ports, and timers. Hence, this architecture is not as integrated as the Hitachi chips, but it uses less power.

3.3.2.6 VLSI Polar IPC/MPC

The Polar chip set consists of two chips. The IPC (Integrated Processor Controller) consists of a 32-bit CPU based on the Intel 80386 architecture and adds power management. The MPC (Multiple Peripheral Controller) is used for the IPC's I/O functions. They are tied together via a 16-bit multiplexed bus to which also a PCMCIA controller can be connected.

3.3.2.7 Motorola "Dragon"

The Motorola 68349 "Dragon I" is a 32-bit processor running at 16 MHz with 3.3 V. Together with the core microprocessor (CPU030), on-board system peripherals (DMA, serial communications, memory interface) are integrated.

A companion chip, the "Astro", provides system interface functions, including infra-red, audio, LCD, and touch-screen, multiple PCMCIA slots, and communications interfaces.

Advanced power management allows the system to tune the clock rate and place the processor in a power-down mode where it consumes less than 1 mW (normal consumption is 300 mW at 16 MHz).

This chip set is especially useful for Personal Communicators, because it provides many of the required interfaces and is used in Motorola's *Envoy* Personal Intelligent Communicator (cf. section 7.4.2).

3.3.2.8 PowerPC 602

The PowerPC architecture [Motorola] was jointly developed by IBM, Apple, and Motorola. It is a powerful RISC architecture with future developments in mind (e.g. 64-bit processing). The PPC 602 is a low-power (3.3 V) version of the original PPC 601, a high-performance super-scalar 32-bit architecture. It features a single-precision FPU[24], execution of five instructions per clock cycle (five-stage pipeline), power management (three power saving modes: doze, nap, and sleep, and automatic dynamic power reduction for idle units), and two separate, time-multiplexed memory management units for instructions (32-bit) and data (64-bit) as well as a 4 KB cache. Running at 66 Mhz, the PPC 602 needs 1.2 W during full operation.

The PPC 602 is a very powerful low-power processor, but it does not integrate peripheral controllers. It has been rumored that the PPC 602 might be the processor of choice for Apple's future pen-based systems.

3.4 Batteries

Batteries are vital for mobile systems [Baran 3]. This topic was briefly mentioned when discussing the power supply for low-power CPUs.

24. Double precision floating-point calculations are trapped by software.

There is a difference between regulated (plugged) and unregulated (battery) power supply in the curve of supplied voltage: the ±10% tolerance suitable for regulated power supply is not sufficient for unregulated power. For a mean voltage of 3.3 V at least 3 battery cells are needed, which will start at a total voltage supply of around 4.8 V. During the discharge cycle the voltage drops to about 2.4 V slowing down the overall processing speed by 50%.

The two different types of batteries have slightly different characteristics.

- Primary cells generate energy through an irreversible chemical reaction (alkaline and lithium cells).
- Secondary cells use a reversible chemical reaction, i.e. they are rechargeable (nickel-cadmium (NiCd), nickel-metal hydride (NiMH), and zinc-air cells).

Primary cells have a better energy/weight-ratio than the rechargeable cells and are mostly used for memory backup (SRAM) which does not use much energy. All battery cells start at 1.6 V per unit (lithium cells start at 3 V) and go down to 0.8 V when depleted. Secondary cells usually have lower voltage rates in the long run but higher peak voltages.

The 2:1 discharge ratio can be partially regulated by means of additional control circuits. Six NiCd cells can be connected in series giving a maximum voltage of about 9.6V that is stepped down to 5V for the whole discharge cycle. For mobile use it would be better to connect two primary cells in parallel and step them to yield 3.3V continuously. Of course, there is always a trade-off to be made between size and weight versus battery life.

For the future, there is no such thing as a Perpetuum Mobile and the storage of energy is limited, although e.g. zinc-air batteries are expected to store four times the energy of NiCd batteries, although not without additional weight.

Another difficulty with battery-powered systems is the connection between a fully charged and an almost depleted device and to recognize the high and low logical levels of each system correctly.

3.5 Personal Computer Cards

The standard of the Personal Computer Memory Card International Association (PCMCIA) and the Japan Electronic Industry Development Association (JEIDA) [Alford], [PCMCIA], [Prophet] defines credit card size extension cards for computers. These extension cards are vital for Pen Computing because they can be used in rough environments – there are no moving parts – and the are very small and light. The standardization of such cards is also very important for mobile computing. The initial intention of the PCMCIA/JEIDA was to define a standard for memory cards only and was announced in version 1.0 in May of 1990. Very soon it became clear that the same cards were very useful for other applications, too. Version 2.0 was announced in September, 1991, and is now supported by more than 350 companies worldwide which are members of the PCMCIA.

The standard specifies both physical and software attributes of the cards. There are two different versions and at least 3 different types.

All versions and types so far have two layers of 34 pins (totalling 68). The cards plug into the device interface port (male) via a socket connector (female). The pins providing power are a little longer than the other pins, so that the cards can be inserted and removed while the computer is running. Of the remaining pins 16 are reserved for the data path (hence, it is a 16-bit architecture) and 26 pins/bits are used for addressing (allowing for 64 MBytes of storage per card).

3.5.1 Version 1.0

The first use of PCMCIA cards was for memory extension [Reimer]. Several types of memory were supported in version 1.0:

- ROM
  Read-only memory.
- MROM
  Mask ROM cards contain data that must be burned in at a factory and is read-only from then on.
- OTP or PROM
  One-time programmable ROM. Can be written exactly once and behaves then like ordinary ROM.
- SRAM
  Static random access memory like the kind used in all desktop computers. These cards require a replaceable lithium battery for storage outside of a card slot and have a write protection/enable switch.
- Flash RAM
  Flash RAM [Dipert], [Eldridge] is similar to EEPROM, because it needs no backup batteries and can be erased electrically (EPROMs need ultraviolet light), but it is impossible to erase and rewrite individual bytes, only the entire device or a few large blocks (zones) can be erased.

3.5.2 Version 2.0 (JEIDA 4.1)

There is no restriction on the purpose of a version 2.0 card. These cards may be used for memory extension, (wireless) fax modems, pagers, mini hard disks (type III, currently 100-200 MB), GPS receivers (global positioning system, cf. section 5.6), SCSI and Ethernet/3270/Token Ring adapters and many more.

Version 2.0 introduced XIP (execute in place), a mechanism that allows applications to run directly from the card without using additional RAM (due to relocation) in the computer's memory. Version 1.0 actually supported XIP for ROM cards only. Version 2.0 also allows cards to run on 5.0 V or 3.3 V voltage mode.

3.5.3 Types

The card type defines only its physical size, and different types vary only in the thickness of the cards:

- Type I 54 mm x 85.6 mm x 3.3 mm
- Type II 54 mm x 85.6 mm x 5 mm
- Type III 54 mm x 85.6 mm x 10.5 mm
- Type IV 54 mm x 85.6 mm x 14.5 mm

Originally, only types I and II were officially defined. The thicker types were later used by several manufacturers and adapted by the standard. Computers supporting one type of cards accept thinner cards because the thickness of the cards' border does not vary.

On the software side the standard defines Card Services and Socket Services that allow for easy self-configuration of the host system to a specific card. Socket Services support the sockets that connect the card via adapters to the host system. They feature memory-space windowing and optional error detection code (EDC) generators and a set of functions that give access to the card's settings. Card Services are at a higher layer providing for functions that deal with integral cards or multiple cards, sockets, and other system resources.

A newer version of the standard – now simply called PC Card – adds several features that generate problems with earlier cards:

- The PCMCIA and JEIDA versions of the standard are now identical.
- An expanded Card Information Structure (CIS) is mandatory and allows for automatic configuration in the host computer.
- The cards must not draw more than 100 mA when first inserted and provide access to the CIS information at that power level.
- A fast 32-bit bus has been added.
- Direct memory access (DMA) is now supported by the cards.

It is very positive that the standard was adopted by so many vendors and at such an early stage of the growing Pen Computer market.

3.6 Conclusions

Many of the topics discussed above do not apply only to Pen Computing but to Mobile Computing in general. But for Pen Computing the restrictions are much harder. If, for example, a keyboard is used for input, the size of the device is almost given by the size of the input device. In Pen Computing the size of a device is given by the size of its output device, which is derived from what the user actually needs. Therefore, the human-computer interface of pen-based devices allows for more compactness. At the same time, light-weightiness, computational power, and wireless communication capabilities are required. The components are ready for use, but the integration is difficult and has not reached the desired state, yet. The path to combining advanced computer technology with consumer electronics – TVs, HiFi equipment, telephones, etc. – is nevertheless very seductive and being followed by many developers.

4 Handwriting Recognition

The recognition of handwritten text is an important requirement for Pen Computing but far from a trivial one. Many different initiatives were taken to solve this problem and it is not yet known which is the best and the results are still not perfect.

It is important to stress two points:

- Handwriting recognition is not a new technology[25], but it has not gained public attention until recently.
- The ideal goal of designing a handwriting recognition method with 100% accuracy is illusionary, because even human beings are not able to recognize every handwritten text without any doubt[26], e.g. it happens to most people that they sometimes cannot even read their own notes. There will always be an obligation for the writer to write clearly. [LaLomia] found recognition rates of 97% and higher to be acceptable by most users.

Keeping this in mind, some efforts taken for handwriting recognition (HWX) will be outlined in order to illustrate the difficulties.

A completely different approach will then be presented: *unistrokes* are specially designed characters that are optimized for easy recognition in order to improve the recognition rate.

Finally, two different standards for storing hand-drawn input – called *Jot* and *UNIPEN* – are presented.

4.1 Optical Character Recognition vs. Handwriting Recognition

Optical Character Recognition (OCR) is already well established in commercial applications, e.g. in fax reader programs that can recognize printed text. OCR always starts from a scanned image of a text that was printed mechanically, i.e. a bitmap representation. A series of transformation techniques are applied to this pixel field:

- Template matching is a method for comparing pre-stored patterns of font faces to isolated parts in the image, where the two should match as closely as possible. This approach works only with a limited number of fonts and sizes and is especially susceptible to rotation and distortion.
- Feature extraction and detection are not subject to these problems, because these methods are based on finding features in the image that do not rely on certain positions, sizes, or angles of rotation. Images are often pre-processed, e.g. using line thinning to eliminate variations of different pens. The features that were found are classified[27] and assembled together and then compared to feature sets for characters to match.

For handwriting recognition (HWX), template matching is not suitable, because the writing styles of different people vary too much; even a single person does not always write in exactly the same style. Feature matching, on the other hand, is a very important technique for HWX. The feature matching techniques are almost the same for OCR and static HWX, but there are some differences when dealing with dynamic HWX.

25. The history of on-line handwriting recognition goes back to 1957, when T. L. Dimond presented "Devices for reading handwritten characters", namely his *Stylator*, at the Eastern Computer Conference.

26. Humans are able to recognize about 96% of hand*printed* single characters, i.e. without contextual information [Neisser, Weene].

27. Classification is done by various methods, among which dictionary lookup and decision trees are the most commonly used.

Static HWX describes the minimalist approach of off-line, bitmapped, and printed letter recognition while dynamic HWX is used to describe the idealized approach with on-line, vectorized, cursive, contextual, and self-learning recognition. Both approaches are discussed and compared in the following section.

4.2 User Interface Aspects

The requirements for user interfaces supporting handwriting recognition can easily be deduced from the pen and paper interface metaphor. First, there should be no constraints on what and where the user writes. It should be possible to use any special character commonly used in various languages, the constraint to use only ASCII characters is awkward, especially with the growing internationalization that the world is facing these days. The ideal system is one that supports handwritten input of Unicode[28] characters.

The second requirement is that text can be written along with non-textual input, e.g. graphics, gestures, etc. The recognizer(s) must separate these kinds of input.

It is interesting to take a look at the coincidence of user interface aspects with the implemented HWX algorithms on which they depend directly. In the future, these mutual dependencies should be eliminated, at least to a great extent, in order to make the user interface more flexible and easier to use.

4.2.1 Off-Line vs. On-Line Recognition

Off-line recognition [De Bruyne, Korolnik] is done by a program that analyzes a given text when it is completely entered, hence there is no interaction with the user at this point. On-line recognition, on the other hand, takes place while the user is writing. The recognizer works on small bits of information (characters or words) at a time and the results of the recognition are presented immediately. The nature of on-line systems is that they must be able to respond in real time to a user's action, while off-line systems can take their time to evaluate their input: speed is only a measure of performance, not of quality.

The combinations of on-/off-line with bitmap/vector recognition are discussed in the next paragraph.

4.2.2 Bitmapped vs. Vectorized Recognition

Bitmap representations of handwritten text are typically obtained by scanning a paper original. The methods used for handwritten (handprinted in most cases) recognition of bitmaps are similar to OCR and rely mainly on contour and other feature extractions.

Vectorized representations are usually interpolated from the coordinate points that digitizing tablets provide. Together with the location (coordinates) of the input it is often possible to get additional information, such as the direction of the path, the pressure of the pen towards the tablet, its angle, speed, and acceleration. This information really makes a difference because it is much easier to adapt to a user's writing style.

While bitmap recognition is typically done off-line, on-line recognition usually uses vectorized representations. There is also a need for off-line recognition of vectorized input, namely when the user takes notes storing the original input as so-called digital ink (including all of the additional information from on-line input), with the ability to recognize the digital ink notes later. This capability is referred to as deferred recognition. On-line, bitmapped recognition does not really make sense and is rarely used.

4.2.3 Boxed vs. Free-Form Input

The use of boxes in which the user writes single characters makes the separation of the characters an easy task. For the user it is often uncomfortable to write in boxes, especially because you cannot choose the place where you want to write. The use of input boxes is acceptable for forms, because people are used to this already on paper and the input is limited to a few characters.

The advantage with free-form input is that size and location information can be used to place the recognized text on the display. Apart from that, it is also much easier for the user to write at any place wanted. The recognition is of course much more difficult, because there is no implicit information on the character separation, orientation, or size of the text.

4.2.4 Printed vs. Cursive Writing

Until recently, only the recognition of handprinted writing has been studied, but with the growing wide-spread use of pen-based systems, cursive writing becomes important. Most systems do not yet support the recognition of cursive (script) writing.

Printed characters are much easier to recognize because they are effortlessly separated and the amount of variability is limited. There have even been attempts to standardize handprinting (American National Standardization Institute's (ANSI) OCR handprinting standard, 1974).

For users it is most uncomfortable to use such standards, because they will always try to return to their own writing style after a short time. This results in sloppily written printed characters or pseudo-cursive writing. Still, the separation may be easier although many people tend to write in a mixed printed/cursive style, where they connect combinations of letters that appear often together (e.g. ends of words like "–er", "–es", "–ing").

4.2.5 Recognition of Letters vs. Words

The recognition of cursive writing is almost impossible without the use of vectorized input, and it is easier to recognize whole words than single characters, because the difficulty in separating the characters is less. The recognition of words assumes the existence of a dictionary, because only words can be recognized that are known. Exactly this limits the usability of dictionaries: they are inevitable, but there will never be a single dictionary containing all the words that a person would

[28] Unicode is a standardized character set that uses 16-bit codes (7-bit ASCII characters are a subset thereof) to encode virtually all the characters of all languages currently used all over the world.

want to write, let alone the problems with proper names or storage capacities. Customizable dictionaries are one solution, but there will still be a need for an additional way of entering words letter by letter.

4.2.6 Contextual Information

Just as dictionaries limit the input space by allowing only a subset of words, there are also ways of limiting the input in a certain semantic context. On the character level, the context can be set to the word that contains it. For example, "O"/"0" or "1"/"l"/"I" look the same, possibly even in the vectorized form, but it is less probable to find digits in the middle of words, than together with other digits (except for programmers). The same is true for the use of upper and lower case characters. Again, this method can be supported by a dictionary.

There are also restrictions on the semantic context on the level of input fields. If, for example, in a certain field only numbers are allowed for input, there is no need for accepting words. It is also useful to allow subsets of dictionaries, e.g. for an input field restricted to accept country names. The accuracy rises tremendously with the use of such limited input spaces, but they are application specific and are not tolerable for general-purpose note-taking tools, for example.

4.2.7 Trainable Recognition

Because every writer has a personal writing style, there is no globally definable set of handwritten text. There are some basic forms, though, that everybody is accustomed to (learned at school, probably) and that are required for making a text readable by other people. From that initial set of variations, there should be a user-specific definition of writing style. One way is to provide samples of one's handwriting to the recognizer, another way is to let the user select shapes from a given list. Both ways are fine, but only as a set-up. The user will not continuously write in the same style over a longer period of time and often the provided samples are different from the writing when the user cares less about writing style.

Therefore, the recognizer should be able to adapt to the user's handwriting style and continue to learn while the user is writing. The recognizer can learn, for example, whenever the user corrects a misrecognized word. The learning may result in resetting the initial set-up to match the real style more closely than the user specified, or weights may be reset, or new style templates added.

Continuous learning guarantees that the recognition does not get worse when the user changes writing style over time – and this does happen.

4.2.8 Mixing with Graphics and Gestures

With on-line recognition there are many types of data that are entered, not only text. Because the pen – ideally – is the only input device, a user must also be able to enter non-textual data, e.g. graphics and drawings with it. And it must provide a way for issuing commands like gestures, selecting menu items or icons.

There may be different recognizers for each of these data types, but they all share the same problem of separating the input. If the user, for example, draws a circle on the display, the intention may be to leave it as is (a drawing), recognize it as a circle (a graphical object), as a "0", or as an "O", or as an Edit gesture[29], etc. The information of the semantic context may be used to sort this out. Some systems, however, require the user to set the appropriate mode, which is uncomfortable.

4.3 Implementation Approaches

The development of algorithms for handwriting recognition is still not in a consolidated state. Many people worldwide work on various approaches to solve this problem, especially since the research has been publicly followed with interest recently.

The first steps were made in the pattern recognition community and only later the computer graphics people caught interest. This is essentially the case because almost all the methods used over time are applications of knowledge-based[30] principles and techniques. Some techniques are similarly used in speech recognition. One major problem beyond the recognition itself is to find a resolution strategy if more than one recognition engine are involved, e.g. for text (letter-by-letter and word-based), graphics, etc. An algorithmic description of handwriting seems to be very difficult. Therefore, stochastic methods, which require much example data for training, are used in most systems. Neural Networks, Fuzzy Logic, and Hidden Markov Models are covered in the following sections.

Unfortunately, an in-depth insight to all the methods that are currently used is beyond the scope of this work[31], but one can say that there are successful implementations commercially available[32] and that research is now widely supported with more and more people getting involved.

4.3.1 Neural Networks

Neural Networks [Krzyzak, Dai, Suen] simulate the signal propagation found in human brains. Brain cells are connected to each other in a network structure and each cell has a varying activation level. The current activation level is determined by the activation signal a cell receives from other cells via its synapses (connection points). The same is true for synthetic neural networks, where nodes are connected. The signal transmission is controlled by activation functions. The layout of a neural network is very important. Typically, a layered structure is chosen: an input layer receives the preprocessed input information and an output layer encodes the results in the activation of the output nodes. In-between, any number of layers are possible.

The most remarkable feature of the network lies in its ability to learn by means of adapting its activations functions in order to match the input patterns to output patterns.

29. Used in PenPoint.
30. Once called Artificial Intelligence (AI).
31. Cf. [Suen, Berthod, Mori], [Tappert, Suen, Wahakara], and [Impedovo, Simon] for detailed overviews.
32. The problem with accuracy rates will not be touched here, because there are no globally applicable measures and the expectations are too different.

In handwriting recognition, the input nodes are presented a vector of features that were extracted from the drawing.

Apart from the network's layout, the pre-processing required for this input vector is the most difficult task. The feature extraction should be invariant to scaling and rotation, e.g. using Fourier descriptors. Timed signals provide additional information, be it in the form of stroke sequences or building a context window to minimize letter separation.

The output layer consists of a vector of nodes, each of which corresponds to a recognizable character. The activation of each output node equals the probability for the corresponding character to match with the drawing.

Initially, neural networks are trained in an extensive learning phase and the learning may be continued while the network is used. The effort lies in the learning and not in extensive programming. Therefore, neural networks are very suitable for handwriting recognition. On the other hand, the restriction of the output layer to recognize exactly the characters represented by its nodes makes word recognition difficult, for example.

4.3.2 Fuzzy Logic

Fuzzy sets [Chang, Pavlidis], [Siy, Chen], in contrast to conventional set theory, allow their members to belong to them partially. The membership function defines how much an element belongs to it. Typically, a numerical grading that ranges from 0 to 1 is used. The fuzziness in handwriting comes from the variability of human drawing, hence the two fit very well together.

The features that are extracted from the drawing (e.g. line segment types, represented by a graph) are compared to fuzzy variables. Fuzzy variables are initially defined for each character in an abstract way in the prototype memory (e.g. "A" consists of three straight lines, two of which cross at the top...). The comparison itself is also fuzzy, i.e. by membership grade, and yields a membership distribution of the recognizable characters. Afterwards, the process of defuzzification results in a guess for the recognized character.

The fuzzy logic approach works well with handwriting because the nature of handwriting is very fuzzy indeed.

4.3.3 Hidden Markov Models

Hidden Markov Models (HMM) are interesting especially for on-line cursive handwriting recognition (as well as speech recognition [Huang, Ariki, Jack]). The principle is based on a stochastic model for graph representations of hand drawn characters and words. The digital ink is converted to a graph, where the nodes reflect changes in direction (e.g. using Freeman chain code). The graph includes information for time as (x(t), y(t)). The HMM is built on a matrix (equal to a fully connected graph) of transition probabilities. In a word-based recognizer, the Markov matrix is initialized for each word of a dictionary (learning phase). In the recognition phase, the input is compared to the matrix according to transition probabilities for each node yielding a recognition rate for words. The recognition of entire words facilitates the separation, but a large model dictionary is required. With a pre-segmented approach parts of words are recognized with the Markov matrix and finally compared to dictionary entries. This is a practical approach because writers write words in parts, too.

4.4 Unistrokes

Instead of dealing with all the problems of handwritten text, one could also make life easier for the recognizers by forcing the user to write in a predefined manner which allows only little variation. Unistrokes are simple, easily recognizable characters.

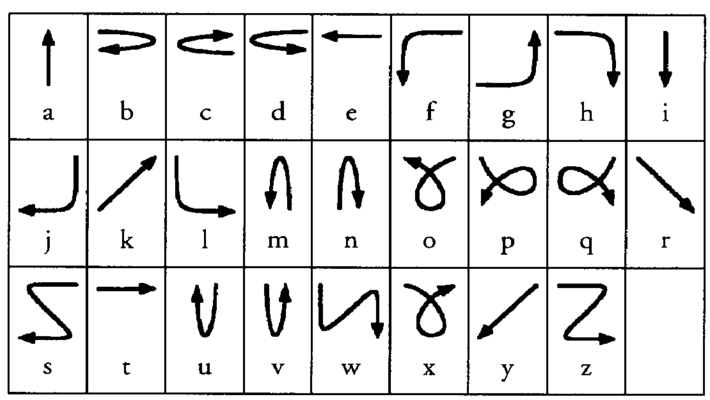
The idea behind unistrokes is to provide a mode for handwriting based on shorthand systems. The design proposed by [Goldberg, Richardson] is based on two abstractions:

- each character of the unistroke alphabet is drawn with a single stroke (hence the name) and
- the characters are maximally spread over the sloppiness space, i.e. they are easily distinguishable even with some inaccurate drawing.

The separation of individual characters becomes trivial. Words are separated by a single tap of the pen. This results in much higher recognition rates and higher speed. It is even possible to write the characters on top of each other to save space and movements by the hand and there is no need for looking at the text while writing. The system works with the same settings for every users.

Of course, the user must learn the alphabet, but it seems that this takes only a short time with the proposed alphabet. [Goldberg, Richardson] match one unistroke character to each letter of the Roman alphabet without separating upper and lower case characters[33] (cf. table 3).

Table 3: Unistrokes for the 26 characters of the latin alphabet



The shapes are derived from 5 basic strokes, each of which can be drawn with 4 orientations and in 2 directions, totalling 40 possible variations. This character set is sufficient for the simple

33. Upper case characters may be entered by pressing a button on the pen or by tapping an icon, depending on the design of the digitizer.

alphabet assumed, but for a real-world application, the minimal character set must exceed the 7-bit ASCII set (127 characters, including control characters). This alphabet may be improved by using a phonetic alphabet, rather than one that matches the letters of the Roman alphabet, but it would be harder to learn.

It is at least questionable whether a user would and should change writing style from the one used on paper. From what is assumed and illustrated throughout this text, it can be followed that a user interface that does not accept regular handwriting is not acceptable.

Based on the work on Unistrokes a commercial variation – named *Graffiti* by Palm Computing – has become publicly available for several pen-based devices. Graffiti is not strictly based on single strokes and, therefore, allows for the input of all common characters, including accents, etc. The characters must be written in a small box inside a window on the display. While many users of Graffiti are very happy with the near-100% accuracy, the solution of using a new alphabet should only be temporary. Hopefully, research will bring forth much improved HWX engines in the near future – provided there is enough support to do so.

4.5 Storing Jotted Notes

Sometimes, it is not necessary to recognize text immediately. A user might want to take notes, e.g. on the telephone, and later come back to those notes for recognition (deferred translation) or editing. The user may also want to let the scribbles be what they are. In any case it is important that the input can be shared with other users using different computer systems and that all the properties of the original input are still accessible. Deferred translation is useless if only a bitmap of the original handwriting is retained. The representation of a user's pen input is called digital ink. If all information is contained in the digital ink, it would be even possible to search uninterpreted text, with the translation taking place at search time.

Jot [Apiki], [Bricklin], [Software Publisher's Association] is a definition of digital ink that provides all of the features mentioned above. The importance of digital ink to be interchanged between different systems is taken care of by the fact that Jot was defined as a standard by the leading companies in Pen Computing[34]. Jot is independent of a specific application or platform format.

The two main goals of Jot – storage for reproduction, deferred translation, etc. and interchange – are possible with the open architecture of Jot. Applications that process Jot data may read only the attributes they are interested in, leaving out others, and the attributes are extensible, leaving room for new ink attributes currently not thought of. In short, the goals for Jot are: simplicity, compactness, compression, inclusiveness, compatibility, and expandability.

The ink is stored in vectorized form (important for resizing and representation on higher resolution output devices) and currently may include the following attributes[35]: stroke order, bounding coordinates, groups of strokes, timing, pressure, stylus angle (in x and y directions), pen color (including opacity), scaling and offset, height over the digitizer surface, buttons on the pen, and pen tip type. Speed and pressure, for example, are important for signature verification, and color may be used for user identification in a multi-user environment.

Attributes that are not used do not use up storage space and there is even an option for automatic compression. The individual strokes are stored as ink bundles in which the strokes are sorted according to time.

The specification of Jot is written in C and consists of a series of constant and type definitions, primarily using a variable record. The INK_BUNDLE_RECORD and INK_END_RECORD build the central record structure between which any number of additional information records can be inserted. Several such structures (bundles of strokes) can make up a stream (which equals a complete drawing).

Standardization of interchangeable data is important. Fortunately, Jot was defined at an early state of the still emerging Pen Computing market. Unfortunately, Slat , Corp. – Jot's major supporter and developer – has ceased to exist which leaves Jot in an uncertain situation.

4.6 Exchanging Handwritten Text Information

While Jot is a binary data format aimed at exchanging any kind of pen-generated input, or electronic ink, efficiently, developers of handwriting recognition engines have slightly different needs. The International Association of Pattern Recognition (IAPR) defined UNIPEN [Guyon, Schomaker, Plamondon, Liberman, Janet] as an extensible framework for pen information – text, signatures, gestures – with the following characteristics:

- Portability.
  UNIPEN is an ASCII format which is easily readable on any machine.
- Human Intelligibility.
  The keywords are easily understandable English words.
- Authentic.
  The UNIPEN format retains as much information as possible from the time of the generation of the pen information. Additionally, there is a easy way for adding annotations to the source data.
- Comparability.
  Comparability may be the most important feature for HWX developers, because it provides the means for testing a HWX engine based on a large collection of samples and therefore set up independent benchmarks.
- Extensibility.
  The minimal required information is a stream of x and y coordinates, which may be grouped as a set of components. Any number of keywords may contain additional informa-

[34] Slate, Lotus, GO, Microsoft, Apple, General Magic, and others.

[35] The attributes that are contained in a Jot data file depend on the digitizer technology used. A digitizer that records only pressure (passive pen) cannot determine the angle at which the stylus is held relative to the digitizer board.

tion.

The IAPR is currently setting up a data base of handwritten samples in the UNIPEN format that should be publicly usable. These samples include cursive, hand-printed, and mixed characters, words, and sentences. All of the important players in the Pen Computing field are contributing their data and participation is open. For the moment, only the Latin alphabet is used as input, but UNIPEN is open for other alphabets as well.

The expectations of the IAPR regarding UNIPEN are to bring research in HWX closer together and help develop better recognition engines in the near future.

4.7 Conclusions

Handwriting recognition is a difficult task and there is currently no single solution that matches all the demands that were outlined in this section. Nonetheless, the results are promising and the possibilities are seductive. With the renewed interest in this field, also outside of the traditionally devoted Japanese and Chinese research laboratories, handwriting recognition will be generally used in the nearer future. The methods that have been described and others can be combined for better results. For example, [Schenkel, Guyon, Henderson] have successfully implemented an on-line recognizer for cursive script using a hybrid system with Neural Networks for feature extraction on the letter level and Hidden Markov Models for word segmentation. Especially HMM systems that rely on statistical data will benefit from the continuing research that provides more exact data on how people write.

5 Mobility

Pen-based systems are convenient because of their ease of use. The use of a pen instead of a keyboard allows pen-based devices to be much smaller and more portable. With portability comes the wish for mobile, wireless communication, because it is increasingly important to have constant access to various electronic information and to be reachable and mobile at the same time. Because pen-based systems can be used at any place, the combination with wireless communications is very seducing. Pen Computing and wireless communications make up a powerful and important couple.

Mobile, wireless communications face several problems compared to stationary, land-based communications [Arnbak], [Bohländer, Gora], [Feuerstein, Rappaport], [Forman, Zahorjan], [Imielinsky, Badrinath], [Mello, Wayner]:

- The environment interacts with signals. Hence, there are more disturbances (noise, echoes).
- There are more disconnections. The mobile systems must handle frequent disconnections and reconnections, because of the lower reliability of connections.
- The bandwidth is lower (by a factor of 1000–100000[36]), i.e. less information can be transmitted per time.
- The higher error rates further delay communication, because additional error checks have to be included and information has to be retransmitted more often.
- The number of devices in a network varies, e.g. at gatherings, where a local concentration of mobile users occurs. This affects the available bandwidth.
- Many networks together form a heterogeneous (meta-) network. When moving from place to place, the user gets in contact with different networks to which the device must be able to adapt.
- Users are reached by calling their numbers. Because the users are mobile their addresses change (moving targets), especially if they move between networks.
- The information a user requires may depend on the location, e.g. to find the nearest doctor.
- Security in wireless communications systems is more difficult to achieve, because there is no hardware to connect to – all information is in the air.

Despite all the difficulties that mobile, wireless communication systems have to deal with, these systems are very important and asked for. The reason for the interest is that these systems alone provide the means for increased mobility without loosing the connection with other people, even if it is only for wireless phones and faxes, or electronic mail.

The following sections provide an overview of various mobile systems: connection of mobile devices to stationary systems, pagers, indoor and outdoor wireless phones, cellular systems, and satellites. This is followed by a section on the Global Positioning System (GPS) and a section concerning distributed information processing.

5.1 Connection to Stationary Systems

Currently, wide-spread use is made of portable computers (notebooks) that are temporarily connected to desktop computers, e.g. at the office. The simplest way of doing this is by using a tether, such as a serial or parallel cable and software that up- and downloads (synchronizes) the data on both systems. Mostly, the portable and stationary computer systems use the same file and operating system. On the other hand, there is no need for such a restriction if the data can be converted to be used on either platform even if, for example, one system uses files and the other system uses a persistent object store.

A more convenient way of linking two systems temporarily together is by using Infrared (IR) connections[37]. The connection is obsolete when the stationary system is not a computer system on its own but only a docking station for the portable computer. The docking station provides for peripherals that are not mobile, e.g. large displays, hard disks, and printers. The portable computer is simply inserted into the docking station.

A sophisticated solution are wireless Local Area Networks (LAN). Wireless LANs connect stationary and portable com-

36. Mobile data transmission 10 kb/s
Ethernet 10 Mb/s
ATM (approaching) 1 Gb/s

37. The Infrared Data Association (IrDA) was set up by the major companies in order to develop a standard for IR communications among a wide variety of devices. This standard is currently being implemented by most of the Pen Computing contenders (cf. PDA Developers, Vol. 2.6, Nov/Dec 1994).

puters by radio signals, usually inside a building. A wireless LAN can detect when an employee returns to the office and automatically synchronize the data brought back from an outside trip.

5.2 Pagers

Pagers are messaging devices that allow for one-way communications (receive-only) using radio transmission that covers large geographic areas.

Several types are distinguished:

- Voice Paging
- Tone Only Paging (beepers)
- Display Paging (text messages)
- Value Added Services (stock exchange, airline information, etc.)

Following an international standard, pagers will operate at 930 MHz in the future, although there are currently many different systems around.

ERMES, the European Radio Messaging System, that operates at 169.6–169.8 MHz with a transmission rate of 6250 bit/s was defined by the European Telecommunications Standards Institute (ETSI) in 1986.

In the U.S., mainly two paging systems exist: *SkyTel* and Motorola's *EMBARC* (Electronic Mail Broadcast to a Roaming Computer). SkyTel is the largest paging service in the U.S., but EMBARC is technically more advanced. SkyTel transmits only 7-bit messages of up to 240 characters while EMBARC transmits 8-bit files up to 1500 characters in order to allow transmission of binary files, too. EMBARC is based on the X.400 standard which makes it open for almost any e-mail system.

5.3 Digital European Cordless Telecommunication (DECT)

DECT is a European standard for digital wireless indoor communications. Users of a central in-house communication system can be reached within a radius of 200 m inside buildings. DECT operates at frequencies of 1880–1990 MHz divided into 10 carrier frequencies using FDMA[38] (frequency division multiplex access) each of which is divided into 12 channels using TDMA[39] with 10 ms time slots. Together, this yields 120 duplex channels. The transfer rate of 1152 kbit/s for each channel is sufficient for data communications, too (wireless LANs).

5.4 Telepoint

Telepoint is a notion for various systems that provide non-cellular (cf. below) cordless pay phone services. They are restricted to one-way communications (send-only), however, and are accessible only in small highly frequented areas. Telepoint, therefore, can only be described as being partially mobile. Several Telepoint systems are already installed around the globe. In the U.S., *RAM Mobile Data* and *ARDIS* are the most popular.

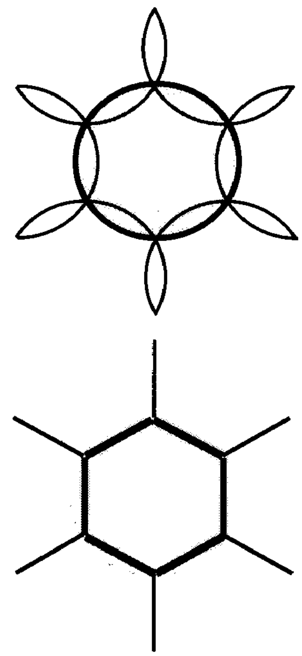
5.5 Cellular Communication

In the beginning, non-cellular communications systems operated at 150 MHz, later at 450 MHz. These were analog systems (NMT A and B networks, in Europe) with overlapping frequency ranges due to the centric radiation of electromagnetic waves. Still analog, the C network was the first type which used cells.

With cells, there are clear boundaries between stations with the same frequency ranges. The separation is possible with some technical effort and management between cells. The cells form a honeycomb pattern. The advantage that results from clear boundaries is that frequencies can be reused in adjacent cells (they do not overlap) and therefore allow for more simultaneous users or more data to be transmitted.

The size of cells varies: smaller cells allow for increased frequency reuse yielding higher capacity, but require more infrastructure. Micro-cells, for example, have ranges of 100 m only. They are difficult to place efficiently because of reflections from buildings. On the other hand, satellites cover thousands of square kilometers and are less numerous at the cost of efficient frequency reuse.

Later networks used in Europe were the D network operating at 900 MHz (NMT 900). The first digital network to be introduced was GSM, operating at the same frequency range. Further developments are heading towards using the 1800 MHz range for Personal Communication Services.



5.5.1 Global System Mobile (GSM)

The Groupe Spéciale Mobile (GSM) [Rahnema] was founded in 1982 as a French/German cooperation for defining a new

38. Frequency Division Multiplex Access: a frequency range is divided into subranges.
39. Time Division Multiplex Access: a carrier frequency is divided into time slots. Each time slot is assigned to a channel

tele-communications standard. In 1989, many more European countries joined to make it the pan-European standard which was then renamed to Global System Mobile.

GSM is the first fully digital mobile standard, i.e. signalisation and transmission are both digital, and it is used for voice and data communications. The user is identified by a chip card, rather than by an end-user terminal. Hence, a user can move with the chip card alone and use any terminal as if it were a personal one.

Two frequency ranges are reserved for GSM: 890–915 MHz for signals and 935–960 MHz for the actual communication between mobile terminals and base stations. The communication range is subdivided into channels of 200 kHz each. This allows for 124 channels per cell. Each cell needs a terrestrial stationary base station (BS). The number of base stations and thus the cell size is variable, allowing for higher density in populated areas. Several base stations are linked to a mobile switching center (MSC) via land-based communications.

Three different modes are distinguished when a terminal crosses a cell boundary.

- Hand-Over
  A terminal crosses a cell boundary while in active connection.
- Change-Over
  A terminal crosses a cell boundary without being connected. Still, the location information for the terminal must be updated because it is permanently required for establishing connections. User data is centrally stored in various registers.
- Roaming
  A terminal crosses the boundary between different GSM networks (e.g. countries).

In order to make more efficient use of the available spectrum, the channels are further divided by using time division multiplex access (TDMA) with 8 time slots per channel. One slot takes 0.577 milliseconds and allows 148 bits of information to be transmitted[40]. All channels together can, therefore, transmit 22.8 kbit per second. However, data transmission is restricted to 9.6 kbit/s with the rest left for error checking and synchronization.

A mobile management entity (MME) manages GSM-specific protocols which are not compatible with OSI protocols, although a similar layer structure exists.

GSM network providers offer essential (E) and additional (A) services. Services of both categories are standardized, but only E services are mandatory. Either services are further categorized as:

- Tele Services
  Telephone, messaging, videotext, teletext, facsimile, etc.
- Bearer Services
  Data transmission.
- Supplementary Services
  Closed user groups, user identification, call redirection, conference calls, etc.

Although GSM was primarily developed for voice communications, it can be used for data, too, because it is digital. The low data transmission rate, however, reflects the critical aspects of wireless communications (high error rate).

GSM is a European standard. It has, nonetheless, gained worldwide acceptance and because it is technically superior to current American and Japanese developments, which are still mostly analog, there are high expectations for making GSM a worldwide standard.

5.5.2 Satellites

An inherently global alternative to ground-based communications systems are satellites. They require no terrestrial infrastructure which may be difficult to build up at some locations. The primary scenarios for satellite communications are

- maritime
- aeronautical
- terrestrial (vehicles).

Ground-based communications are more difficult due to much higher reflection causing interference. All of the above mentioned areas should be integrated.

5.5.2.1 International Maritime Satellite Organization (Inmarsat)

Inmarsat was founded in 1979 and has a global communications network operative since 1982. Originally only for maritime use, Inmarsat is the first and still single organization to offer general satellite communication. The user needs a device with an antenna with a radius of 10 cm. There are various Inmarsat systems:

- Inmarsat A
  Primarily for maritime use, this system supports telephone, teletext, fax, data (64kbit/s). It is expensive and requires relatively large equipment.
- Inmarsat C
  A packet-switching system for data communications for general use.
- Inmarsat B, M
  Newer speech and data communication systems.

5.5.2.2 Motorola Iridium

Planned to be operative in 1997, Iridium[41] is a new satellite system for global communications. It consists of 77 satellites at an altitude of 765 km above ground. Each satellite covers an area of 24000 km^2 and is divided into 37 cells. The low altitude of the Iridium satellites allows for small end-user terminals (approximately 500 g).

[40] The 148 data bits are sequentially arranged into 57 bits of data followed by 26 control bits and another 57 data bits, all of which are separated by some additional start and stop bits.

[41] The name is taken from the chemical element which equals the number of satellites with the number of electrons.

The satellites communicate among themselves and with only 20 space-earth gateways at 1610–1625.5 MHz. The gateways are integrated into conventional terrestrial telephone systems. Users will need only one universal access number and be globally routed. The services planned for Iridium include voice, data, facsimile, paging, and positioning.

5.5.3 Personal Communication Services (PCS)

The Personal Communication Services set the direction for future telecommunications interaction techniques. PCS can be understood as an integration of previously described systems. The key word *Personal* implies a shift from today's centricity on addressing end-user terminals (telephones, faxes, etc.) towards addressing people. The difficulty lies in the localization of the people and the goal is mobile reachability. There will be addresses, e.g. numbers, for individuals who are located wherever they are at a certain point of time. Hence, the caller does not need to know where the addressee is located. In addition to this, addressees may also possess multiple addresses for their diverse activities, e.g. for private and business relations.

The prerequisites for such a system are the availability of highly mobile and cheap end-user terminals that combine all sorts of services and a network infrastructure[42] capable of transmitting enough information to satisfy the users.

The identification of a user works by using a chip card, similar to GSM. Identification is required in order to locate users and to account for calls that users can make from any PCS terminal where they identify themselves. Today's unpersonal terminals require the use of pre-paid phone cards or even cash. Also, identification allows a user to access all of the personal settings and preferences at any place. The benefits of preferences are, for example, services like closed user groups (CUG), virtual private networks (NPV), and additional charge services (kiosks).

A distinction is made between discrete and continuous mobility. The former refers to user identification at stationary terminals, while the latter is used to describe the use of portable wireless terminals which are automatically located.

The European Telecommunications Standards Institute (ETSI)[43] defined a standard for a PCS system in 1990 known as Digital Cellular System (DCS) 1800. The number refers to the frequency spectrum used. DCS 1800 is based on GSM 900, with some differences. The first difference is the shift of the spectrum from around 900 MHz to 1800 MHz as well as expanding it from GSM's range of 25 MHz to 75 MHz per direction. The higher frequencies and the lower power[44] (required for cheaper and more mobile terminals) result in a much smaller radius for cells. While GSM's cell radius is around 50 km, DCS will use cells with a maximal radius of 8 km in open environments and 1 km in cities. Microcells with a radius of 150 m and picocells that cover single buildings are also planned. The problem of additional infrastructure needs versus increased capacity of smaller cells has been discussed.

In the U.S., the Code Division Multiple Access (CDMA) method is perceived as the most promising way to PCS [Salmasi]. CDMA was developed by Qualcomm [Salmasi], [Wayner]. It encodes information for up to 10 channels that can be transmitted simultaneously at one frequency. Each channel is assigned a different code that is used to decode the information for this channel only, while the other signals appear as noise. This efficient transmission method is combined with sophisticated hand-over techniques. Currently, CDMA is planned to run at 800 MHz, but will later run at higher frequencies, e.g. 1800 MHz.

Other visions for future telecommunications systems, which are even less detailed than PCS, include the Universal Mobile Telecommunication System (UMTS), Universal Personal Telecommunications (UPT), and the Future Public Land Mobile Telecommunications Services (FPLMTS). UMTS and UPT follow a strategy similar to PCS, but are planned to be inherently global from the start. FPLMTS focuses on the global integration of all kinds of different services in a single system.

5.6 Global Positioning System (GPS)

The Global Positioning System (GPS) is a satellite-based navigation system operated by the U.S. Department of Defence (DoD) [Getting], [Department of Transportation]. It allows a user with a GPS receiver to calculate the currect position on the earth's surface in 3 dimensions (longitude, latitude, and altitude) as well as world time.

The information from positioning systems is very useful for users of pen-based devices, e.g. to adjust country-specific settings such as telephone access numbers, to display location-specific information (e.g. city guides, aeroplane and railroad time-tables), and for updating the time (especially important for mobile communication). GPS receivers are small enough for mobile use – they are available on PCMCIA cards.

The history of positioning systems began with the need for ships to find their way on the sea. The first navigation systems relied on visual contact with land-based light houses. Of course, the lighthouses could be seen only from a limited distance and when the weather was fine. Inertial positioning systems were used to track the route of a moving object unlimited by sight and distance. Unfortunately, inertial systems loose accuracy, because positioning errors increase over time. But a major step forward was the use of radio wave signals in absolute positioning systems. A ship at sea could determine its position (longitude and latitude on the idealized geoid) by measuring the time delays of the signals of two pairs of synchronized wave sources (or three synchronized sources). This system, named *Loran* (Long Range Navigation), was used during the forties. The radio wave signals could be received out to 1200 km by day and 2000 km by night and the system allowed receivers to calculate their position with an accuracy of 1.5 km. The main points of interest in Loran are the use of radio waves, the syn-

42. The PCS network will combine mobile systems with the terrestrial communications infrastructure.
43. The driving force behind this initiative are the telecommunications companies of the United Kingdom. The U.K., among others, is leading the development for the future of telecommunications because this market has been greatly privatized.
44. GSM: 5–10 W, DCS: 250 mW.

chronization of the ground stations, and therefore the passivity of the user who needs only receive the emitted signals and not send out signals making the system secure and usable for an unlimited number of users. These characteristics apply to all later positioning systems.

The first positioning system to introduce the use of satellites was *Transit*, developed by the U.S. Navy. The first Transit satellite was launched in 1957. The land-based radio wave emitting stations were replaced by satellites in order to cover the surface of the entire world. Transit was not very successful because it was only a two-dimensional system and too slow for moving objects, especially for aircraft and spacecraft.

The U.S. Air Force started a project called *Navstar* in 1963 which is widely known today as GPS. Navstar was set up to investigate the requirements for a global positioning system and they found that such a system should ideally

- pinpoint the position of satellites,
- use radio waves that penetrate the ionosphere to guarantee weather independence, and
- measure relative time of arrival of synchronized signals from the satellites.

This ideal was in fact a high tech version of early star-watching techniques.

The GPS system consists of three segments.

- The space segment consists of 24 satellites divided in 6 planes of 4 satellites. Each satellite circles the earth every 12 hours at the height of 20200 km. The signals are sent at two frequencies above 1000 MHz to allow for correction of ionospheric propagation errors. The signals are synchronized by use of atomic clocks aboard each satellite.
- The ground segment controls the satellites and synchronizes the satellites' clocks with a ground-based atomic clock. It also updates the satellites' position information through a series of widely spaced monitor stations whose positions are accurately known.
- The user segment consists of an unlimited number of receivers (one per user) that receive the satellites' signals. Encoded in these signals is information about the position (3D) and time of each satellite. With these four parameters from four satellites the user can calculate the current position on earth and update the clock[45].

For security reasons the information that the satellites transmit is encoded. Two codes (P code for military use and C/A code for civil use) yield different accuracies. Civil users can determine their position with an accuracy of 15 m and update their clock to an accuracy of 100 ns. With differential GPS – using one moving receiver and one static receiver whose position is accurately known – positioning is accurate up to about 1 m. This technique is useful for relatively small areas, e.g. for navigating in the environment of harbors or airports.

45. An equation system of four equations has to be solved for four variables (x, y, z, t).

Apart from the initially intended use in sea, aircraft, and spacecraft navigation[46], GPS will be used for traffic control and car information and guidance systems to prevent heavy traffic and provide drivers with up-to-date information on the situation on the streets. Drivers will be informed about the way to their target including detours and about the scenery they look at while driving.

With the information accessible through the use of GPS two important goals can be achieved: a worldwide mapping grid that eliminates irregularities at country borders due to different mapping schemes and worldwide time synchronization[47] which is important for mobile communication across time zones.

Thus, GPS can help in improved safety and efficiency wherever anything moves. For most uses, though, large and detailed geographical databases are required.

For almost all of the above mentioned applications of GPS, a pen interface is useful, be it for military or civilian use, vehicle control or any kind of location-dependent science.

For example, the California Governor's Office of Emergency Systems (OES) makes use of Apple's Newton MessagePad coupled with a GPS receiver for recording earthquake disaster data faster and more precisely and efficiently. The location and time information is entered automatically and the user adds other information using the pen.

5.7 Distributed Processing & Information Access

Mobile users store information in their terminals. They want to access data of other mobile users as well as information that is provided for larger groups of users, e.g. office information, library systems, shopping, and ticket information. Therefore, the problem of mobile data management arises [Imielinsky, Badrinath], [Rekhter].

The key to mobile information access is to know the location of the desired information. Only, in mobile systems the location changes frequently. Also, the information may be split up in different parts, because there is no central information store. Both problems deal with distributed databases with the added complexity of a constantly changing configuration of information sources. Redundancy cannot be avoided and has even to be included in order to update and query the databases efficiently. The data that a user retrieves as a result of a query in a mobile distributed database depends on the user's location, the location of other users, and the current time. Hence, the information is valuable for a short period of time only. For example, querying for the "route with little traffic from here to the nearest doctor" will result in an answer that is true only as long as

46. Because the GPS is operated by the DoD, its primary *raison d'être* is for military use, e.g. the control of aircraft and missiles during the Gulf War, but the infrastructure is fortunately made publicly accessible at no cost.

47. International Atomic Time is a goal of the International Bureau of Weights and Measures in France.

the location of the user, the location of the doctors and the traffic in the area do not change.

Depending on the query characteristics, location information may be stored at a user's home location server (HLS) or retrieved by use of a broadcast call. The HLS must be permanently updated by the user, e.g. whenever moving from one cell to another. This information may be spread over other server databases that replicate the information (in a flat or hierarchical structure). Broadcasting is useful whenever the addressed user is not explicitly known or when a larger number of users are involved. A broadcast sends out a query to all users in a certain area and relies on the potentially addressed users (or their terminals) to respond. The broadcast may be encoded in order to limit the scope of the addressed users. Broadcasting is also used for reaching an unlimited number of user, much like television and radio broadcasting works. With the HLS method, the main effort for retrieving location information takes place when a cell border is crossed, whereas the broadcasting method requires it to take place at search time.

Partitions are a combination of server databases and broadcasting. Partitions are areas (e.g. groups of cells) where users are often located. The partitions can be obtained from statistics based on a user's communication behavior (location and call probability). A caller can retrieve the partition in which the user to be addressed is located from the HLS, and then broadcast the partition for the exact location.

The information that can be accessed in a mobile distributed database can be

- Private data belongs to a single user who has read and write permissions alone. It may be stored in the mobile device or at a HLS.
- Public data may be read by an unrestricted number of users. Only one user is permitted to update the data.
- Group data can be read and updated by several users.

Group data is the most difficult to handle, because data integrity must be assured with a large number of users willing to update simultaneously. This is even more difficult in a database that is distributed and where users can access from an unpredictable location (variable configuration). Caches can help to recover from the frequent disconnections, but again redundancy is added at the cost of data integrity.

Not only the data, but the processing power can be distributed, too. A mobile device can send a request for processing a computation intensive task to a remote (stationary) computer[48]. It is also possible to use many mobile and stationary devices in parallel, but the cost of communication would probably be relatively high.

The applications that will benefit the most from mobile distributed data storage and processing are *mail-enabled applications*, i.e. applications that let people work together productively over great distances. Remote workgroups are possible this way. Collaborative and ubiquitous computing [Weiser] rely on distributed data. They build the foundation for forms of living and working where people have shared access to data anytime and from anywhere.

48. General Magic's Telescript (cf. section 7.4.1.2) allows for this kind of distributed processing.

5.8 Conclusions

Compared to developments in land-based communications (e.g. time-dependent Multimedia communications with Asynchronous Transfer Mode (ATM)), wireless communications are left behind: much lower bandwidth, higher retrieve cost due to locality of information, variable configurations, and instable transmission quality, etc.

Nonetheless, wireless communications have very interesting areas for applications (integration of wireless telephones with computers, HiFi, video, etc.). The dream of the global wireless Multimedia communication network, however, is still far away. It will not be realized in the next two decades. Efforts are made to deregulate the communications structures in order to increase innovation and affordability. The Federal Communications Commission (FCC) is auctioning 160 MHz of bandwidth for use in PCS by interested and capable companies in the U.S. during 1994. There is no certainty as to whether the commissions charged with ruling the wireless data stream all over the world will manage to define a global standard.

It is, however, important for users of mobile, pen-based systems that their devices work on a global scale, because only this allows for them to be used wherever they go. As soon as the devices are faced to boundaries in relation to their operation they are much less worth. A device might be able to support several communications standards, but true global communications rely on global communications standards.

6 Paradigms

Pen-based computing is different from conventional computing in several aspects because of its higher interactivity, intuitiveness, mobility, distributedness, and integration of diverse media and information. All of these aspects require new paradigms that are covered in the following sections. First, user interface aspects are discussed, followed by a coverage of new programming paradigms. These two aspects are closely related, because the way programmers use to implement user interfaces affects the result. There is, therefore, also a high importance in designing good user interfaces for programmers. The section closes with a categorization of hardware and software found in pen-based systems.

In the following, the term *pen-based machine* will be used for describing a machine that is primarily or exclusively operated by the use of a pen. There have been many systems with support for pen input that used the pen as a secondary input device only, e.g. CAD applications often rely on a digitizing tablet for data entry, but the rest of the program is operated by a mouse. Such a system is not considered as a pen-based machine in this context.

6.1 User Interface

Using a keyboard for data entry requires a user to have two separate tools: the keyboard and the screen. The consistency between the two – or feedback – is represented by a blinking cursor that shows the position of the next character to be entered when the appropriate key is pressed on the keyboard. To move the cursor to another location the user must press the cursor keys that change its position in one of the four directions. Less awkward for positioning is the use of a mouse which follows the movements of the users' hand on a surface. The mouse is also detached from the screen and therefore it, too, needs a cursor representing it on the screen. A mouse cannot be used for text entry.

On the other hand, the pen interface unites text entry and positioning and requires only one integrated device – a digitizing display. Essentially, such a machine appears as a pen-sensitive screen with a built-in computer.

Because the pen is closer to the human body, it also has to be adapted more closely in the interface. Left- and right-handed people need different interfaces, e.g. a close box for a window must be positioned in the bottom left or right corner, so that the display is not covered by the hand.

6.1.1 Notebook Metaphor, Pen and Paper Metaphor

The aim of Pen-based Computing is to imitate a pen and paper interface, e.g. in the form of a notepad or a notebook. The users of pen and paper can organize their drawing and writing without restrictions, only the size of the surface matters but there are no different modes for text and graphic entry, storage and speed problems, or an influence of the input sequence, except for overlapping objects. A notebook also allows users to organize their information in different sections and use a register with handles. Of course, a digital notebook expands the possibilities – e.g. searching through recognized text, cut, copy, and paste texts and graphics, use sound and animation, etc. – but it must not add any limitation to the interaction a user knows from working with pen and paper. This way, even computer illiterate people are able to work with a pen-oriented interface.

6.1.2 Gestures

Gestures are hand movements that are used for signalling through the air. Similarly, another kind of gestures is drawn on paper with a pen, like those used for correcting text – inserting, deleting, etc. of characters, words, and paragraphs. Most of these gestures are widely accepted and intuitive and when used in a pen-oriented environment, the gestures do not only describe the intended changes, but they actually affect the texts and graphics.

Unlike the conventional graphical user interfaces, that require one input device for positioning, another one for entering the text, and eventually yet another one to select the intended command, a pen does all this with a single gesture. A user can, for example, strike through a word in order to delete it, or circle a paragraph to select it for changing its attributes, move it, etc.

[Foley, Wallace, Chan] define six interaction tasks for graphical computer interaction: select, position, orient, path, quantify, text input. All of these can be accomplished simultaneously when using a pen.

A keyboard defines the input text, a mouse is used for positioning it and selecting menu commands. The pen defines the text, its size and location, as well as graphic elements at the same time. There is no two-step interaction, because selections and commands are issued simultaneously. For example, a user can change the style of a part of a text simply by drawing the appropriate gesture over it.

6.1.3 Digital Ink

The cursor that is known from conventional text-based and graphical user interfaces – for marking the position where the next piece of input is placed – is not required in Pen Computing, because the interaction is inherently more direct. The need for a cursor stems from the fact that the input device and the display are detached, which is not the case in most pen-based interfaces. The response from the machine to the actions undertaken by the user by means of the pen are reflected immediately. The trail of the pen is shown as a path on the display known as *digital ink*. This ink can be left as is, converted to text or smooth graphics, or it can represent a gesture for a user's command. In all except the first case, the digital ink trail is removed after recognition.

A difference of digital ink compared to real-world ink is that the pen need not be refilled, and that the same pen can draw fine and bold lines, act as a pen, pencil or brush, in different colors, sizes, etc. – without changing the tool physically.

6.1.4 Recognition of Hand Drawn Input

Recognition of hand drawn input – such as text, graphic elements, mathematical equations, or other technical drawings – is not required for Pen Computing. The use of digital ink alone makes a machine a pen-based machine, too. Especially considering the recognition of handwritten text, some people fear that less than 100% accuracy renders recognition useless. Although nobody would seriously reject a machine with errorless recognition, it is not likely that this can be achieved at all. Not even humans can read each other's handwriting in all circumstances. One way to improve recognition is to allow *deferred translation*, i.e. the digital ink is stored as is at the time of writing and can be translated by a powerful computer later. Deferred translation requires that much more information is stored than needed after immediate translation[49]. Even so, handwriting recognition is currently quite good and it offers many additional interaction capabilities.

Writing down a command that can be executed can be easier than tapping a series of icons or selecting menus in certain circumstances. Written commands are not bound to a certain location on the display. There are written down any place and either executed upon drawing a gesture over them, or when

49. See section 4.5 on JOT for a detailed explanation on exchanging digital ink information.

they are selected and a button is tapped – as is the case with Newton's *Assist* button.

A pen-based machine without recognition of handwritten input may easily become boring, because it cannot deal with the information that was entered, e.g. search an address file for a given name, or reformat and move text and graphic objects.

Compared to pens, keyboards and mice have several problems to get away with:

- Keyboards and mice are not useful for very small as well as for very large displays because they require a fixed position on a surface and use up too much space.
- Keyboards are faster for linear text entry, but in most cases editing of existing text is required which can be mixed with graphics. A single input device that can handle all kinds of data and jump quickly from point to point is easier to use in this case.
- Keyboards are very cumbersome for large alphabets, e.g. Chinese and Japanese (this is why research in handwriting recognition has been more popular in these countries).
- It is difficult for users to write with a mouse because mice are too large to allow for precise finger movements. Because mice record relative positions and movements, it is difficult to align words properly.

A good pen interface should not have any modes for the recognition of handwritten input [50], although input modes simplify the task of recognition significantly.

6.1.5 Document and Data Centricity

Users of pen-based machines do not launch applications – they deal only with their data and documents. Documents are not bound to a certain type of application, because there are no application-specific files. Documents [Reinhardt 3] act as containers for any kind of embedded data[51]. In order to create and edit documents, a user relies on the capabilities of the *tools*, *assistants*, and *agents* that are available on a specific machine (viewing should be possible for any document, even if parts of it were created with tools that are not currently available). Tools are always present and ready for use in generic, i.e. application-independent, documents – they need only be activated and deactivated. Activation implies that the state in which the user left the document is always immediately restored whenever returning to it without explicitly saving the data. A user, hence, opens an empty document – a blank sheet of paper – which acts as a container for the information that is entered with the appropriate tools, often presented in a toolbox window as icons. Switching between different application programs is therefore eliminated. Assistants help user perform complicated tasks that are specified in natural language. They rely on detailed information about a user's habits and preferences in order to handle the requests flexibly and efficiently. Agents activate themselves and perform tasks that a user need not ask for explicitly. Tools, assistants, and agents are explained in detail in the succeeding sections.

In data-centric systems, the raw data is separated from its representation. Multiple documents can be created by assigning various forms to the same data. Data-centricity, therefore, incorporates document-centricity and enhances it by allowing different representations of the same data with the advantage that data that is manipulated in one place can automatically be updated in all other places. This concept is very close the Smalltalk's Model-View-Controller (MVC) concept [Goldberg, Robson]. Data-centricity paves the way from simple application files to live composite objects that can be stored permanently. The Permanent Object Store (POS) is updated whenever the user manipulates data. This data is not associated with any specific application but accessible to all container documents, thus avoiding redundancy. The user needs only pick up the tools that are appropriate for the data to be manipulated and needs not care about when, where, and how the data is stored.

6.1.6 Tools

Unlike the application programs common to current desktop computers, tools for pen-oriented interfaces put the user in a mode for entering data only briefly. Application programs put users in an input mode for the full time they work with them, e.g. a text processor accepts only text. In order to embed graphics into the text, a different application is required and the graphics need to be stored in files in order to import them in the text processor. Tools offer various data manipulation capabilities to the user, but the interaction always happens on the same paper interface. It is just like buying a new pencil for writing on the same notepad, or a pair of compasses for drawing circles. The paper itself does not mind whether text is written on it or graphics are drawn, or mathematical equations, watercolor paintings, etc. The tools make the difference, e.g. one might want to compare a circle drawn by hand and one drawn with a pair of compasses. Digital tools are the same, only more powerful. In this perspective, tools are *passive* and can be changed on the fly, while application programs are *actively* restricting the user's capabilities.

6.1.7 Assistants

Assistants provide an easy way for specifying what should be done in a form that suits the user. Assistants may understand written or spoken commands or graphical gestures and interpret them. Interpreting here means that the assistant can invoke complex actions on behalf of only short commands. A strong requirement for assistants is that they must be very flexible in the form in which the receive their instructions. The command word or gestures and the parameters are mixed and can vary widely from use to use and from user to user. Assistants are in many cases more flexible than menus and macros because the user specifies only what is wanted to be done, instead of commanding every single step. Assistants, therefore, need much background information on typical user interaction (common sense) and should also be able to learn from the user (some kind of intelligence could then be attributed to

50. PenPoint uses 4 phase modes for handwriting recognition (write, recognize, correct, accept).
Newton is more intuitive in text entry and editing (no window opens), but there are separate recognizers selectable (cursive writing, letter-by-letter entry, and graphics); if all recognizers are selected the interface is modeless but less accurate.

51. See [Reinhardt 2] and section 6.2.1 for information on *OpenDoc* and *OLE*.

them). The user activates an assistant by selecting the commands and tapping an assistance button, or by drawing a gesture over them.

6.1.8 Agents

Agents [Miley], [Minsky], [Nagel], [Riecken], [Wayner 4] are even less intrusive than assistants, they work in the background and act on their own initiative, whenever they find information that is relevant to the user. Specialized agents may be added to existing environments or make up integral parts of operating systems.

Characteristics of agents are that they are

- *autonomous*, i.e. they work in the background and are not explicitly invoked (they observe the user and accessible information sources),
- *intelligent*, i.e. they act on their own initiative and can work in heterogeneous environments adapting to manifold situations (they do not necessarily use the same resolution strategy every time),
- *personal*, i.e. they adapt to and learn from their user, and do not stick to a certain solution if the user decides differently (user feedback is an integral part of intelligent agents[52]). Agents can make decisions in situations where it is uncomfortable for the user to be disturbed, but they must never force their user to a certain behavior.

Implementing agents is a difficult task, which may succeed by applying object-oriented programming to knowledge-based and machine-learning techniques (e.g. expert systems and neural networks). These techniques are especially important for modelling real-world environments, or common sense. Only this ability will render software agents useful for technically unenthusiastic users.

Agents are the way in which the increasing information sources can be used effectively, considering the complexity of the new information systems. The idea of using agents is not this new. A decade ago, Alan Kay, Marvin Minsky, and others already worked on them, but today they become a real necessity with the growing worldwide interconnections. Communications is a major application for agents at General Magic's[53] where Bill Atkinson describes them as "controlled, well-behaving viruses". Like viruses, agents can clone themselves, i.e. create other agents if they see a need[54]. They also fit well in the up-to-date client/server architectures, where agents could be running on a remote server independent of the user's present work.

With the use of agents, users are freed from many routine tasks – such as making backups, or searching for news on certain topics – avoiding error sources that come as a result of the natural laziness towards the non-human needs of a computer. Agents should make computers usable and useful for technically indifferent people, too.

A major field of action for agents is information retrieval. Even with locally stored data it is often not easy to find the right information, because the data is stored in various formats and media[55]. Putting together the matching pieces is far from trivial and takes up a lot of time. With the forthcoming possibilities of worldwide communications, the efforts grow exponentially. It can be foreseen that in the future people will not be able to keep up with the information they need and keep track of their appointments without the help of intelligent autonomous personal agents.

One requirement that tools, assistants, and agents have in common is multi-tasking, because they are modeless and/or work in the background. Table 4 compares these three levels of user support with traditional application programs.

Table 4: Comparison of human-machine interaction techniques and user support levels.

Technique/ Property	interactive	data-centric	intelligent	autonomous
Application Programs	•			
Tools	•	•		
Assistants	•	•	•	
Agents		•	•	•

6.1.9 No Computers

Pen-based devices are built on the idea that as little background information as possible is needed to operate them. The knowledge involved should reflect the user's domain and not the computer's. If this goal is achieved, can this low-impact, unobtrusive technology still be called a computer? Not really!

Because other devices that people have become accustomed to are often not less computerized – e.g. HiFi equipment, washing machines, TV sets, pagers, or credit card automata – and are still not recognized as computers, the pen-based devices are electronic tools that are intuitively used – but they should not have to be looked at as computers. The pen and paper interface that is used may be perceived as *Interactive Paper*[56], very much like an extremely featured notebook. [Weiser] coined the term *Ubiquitous Computing* to describe the accessibility of hidden computing power at any place.

52. The term *conditional intelligence* is used to describe this ability.
53. See later for details on General Magic and their *Telescript*.
54. E.g. one agent might create another agent in a node of a network to wait for a certain condition or point of time to invoke an action while the cloning agent itself can go on with its own work somewhere else in the network.
55. Including text, graphics, animation, sound, video, etc.
56. The author uses the term *Interactive Paper* to describe the conceptual capabilities of writing-oriented interfaces, including all sorts of multi-media extensions.

6.2 Programming

The ability to adapt a pen-based device to a user's specific needs is vital. Software is always the main factor in the decision of a user for a certain hardware and/or operating system platform. Adaptation may range from setting preferences to programming macros or self-contained applications. Ease of use and developer efficiency are more important than compatibility with desktop machines. Compatibility is only important for the communication of data. Many efforts are therefore being made to prevent the users from a chaos of new and modern, but isolated developments. The field is still young and it is time now to shape the technology of the future with the best techniques known without the urgency for compromise.

Programming a pen-based device can be done either on the device itself or on a desktop machine. Small, hand-held devices may not be ideal for writing voluminous source code. But programming need not always be based on textual declarations. Visual programming could let a user graphically specify some tasks to be done. The following paragraphs will point out the specific requirements and opportunities that become imminent when programming for pen-based devices.

6.2.1 Object-Oriented Programming

Object-oriented programming (OOP) [Baran 6] has become well established for graphical user interfaces. Pen-oriented interfaces are inherently more graphical than keyboard interfaces and mouse interfaces due to the much higher degree in immediate interaction and the use of digital ink instead of a relatively static cursor.

The opportunity must be taken to apply object-oriented paradigms to the new operating systems from ground up. Object-oriented frameworks build a strong foundation for today's programming needs. A framework consists of a collection of classes or prototypes[57] that implement all the services of the operating system and the components of the user interface in a coherent way, except for the application specific tasks on which the developer concentrates effort. The developer composes and inherits from the framework's components, extending and adapting it to a user's needs. A clever framework is a powerful and efficient tool.

Object-oriented techniques are the ideal implementation of call-back mechanisms and make the bulky event loops obsolete.

Objects are used to represent the data and are therefore in the centre of interest when it comes to data storage and information interchange. Objects provide a clean interface for interaction by use of methods that make their internal implementation exchangeable. The objects can be addressed by sending them messages to which they respond in an internally predefined manner. Thus, it is possible to address remote objects across networks bothering neither the caller nor the called. With this object communication the objects of the whole world will be shared in the future by means of standardized access mechanisms for permanently stored objects[58], creating sort of an Object Universe.

The Common Object Request Broker Architecture (CORBA) by the Object Management Group[59] (OMG) is an attempt heading in this direction. It provides for portable distributed objects. Many contenders in Pen Computing are also putting efforts into object-oriented technology: IBM's Distributed System Object Model (DSOM), NeXT's Portable Distributed Objects (PDO)[60], OpenDoc from Component Integration Laboratories (CIL)[61] [Apple 2], [CIL], [Piersol] or Microsoft's Component Object Model (COM) [Udell 2]. DSOM and PDO are general models for distributed objects, while OpenDoc concentrates on portable documents and data and supports the development of tool-oriented (i.e. data-centric) environments by means of components. These three are based on CORBA. COM, on the other hand, is less consequent, as it does not support inheritance. Nonetheless, in the preliminary form of Object Linking and Embedding (OLE), COM is already available on the Windows platform. Taligent, the Apple/IBM startup company, is developing an object-oriented operating system based on frameworks – called "People, Places and Things" – that are centered on user activities, work flows, and documents rather than isolated applications.

The two currently important object-oriented operating systems for Pen Computing are GO's PenPoint and Apple's Newton architecture. They are covered in detail in sections 7.2 and 7.3, respectively.

6.2.2 End-User Programming

The kind of ultra-personal computers discussed here requires a computer to adapt closely to its user. The user must be provided with a way to personalize and "program" the computer. Of course, a user must not need to use C routines to modify the operating system. Rather, it should be easy to specify needs on a high abstraction level that matches with the application domain.

The user interface must provide two symmetric capabilities:

- Read, i.e. use a computer to get access to desired information.
- Write, i.e. produce information for the user and for others.

If these two functions are provided in a coherent and intuitive way, information can flow between users, and the computer becomes a means of content-centric information production

57. Prototype-based languages – such as *Self* and *NewtonScript* – are discussed in section 7.3.1.

58. A Permanent Object Store (POS) allows object data to be kept alive for an unlimited time, e.g. by using object-oriented databases.

59. The Object Management Group (OMG) is a joint effort of Hewlett-Packard, Sun Microsystems, Digital Equipment Corporation, and many others.

60. NeXT's PDOs are being developed for many platforms, e.g. HP-UX, Sun Solaris and OpenStep, and other Unix platforms.

61. CIL was founded by Apple, WordPerfect, Novell, Borland, and IBM. OpenDoc provides an object-oriented framework for compound documents, i.e. documents consisting of various parts (text, graphics, pictures, sound, movies, etc.) that are integrated. Each part of a compound document possesses its own editor. OpenDoc uses frames to implement this model.

and consumption. Ideally, this will happen worldwide with the help of wireless communications.

6.2.2.1 Forms-based Programming

Forms are widely used wherever data is collected: in hospitals, service people, agents, etc. Forms consist of a set of simple input fields which are laid out in a specific order. Each filed has a very small input domain, i.e. it is well known what data can be entered. Therefore, forms need not rely on difficult free-form handwriting or graphic recognizers, but use radio buttons, pop-up list, etc. instead. Forms are easy to set up and easy to use. Forms generators are already well established and coupled with more or less sophisticated data bases.

6.2.2.2 Hypermedia

A more challenging approach towards user programming is the creation of hypertexts and hypermedia. Hypertexts consist of text into which links are embedded. Links connect one word in a place of the text with other places (typically one) in the text. The links are followed when the user clicks or taps the linked words. The related text portion is then displayed, either in a new window or in place of the previously visible text.

Hypermedia extend this concept to combine linkable graphics, images, video, sound, etc. with the text. A special form of hypermedia are known as *electronic books*. Electronic books follow the structure of paper-bound books: chapters, table of contents, etc. and add the capability of hypermedia links, electronic bookmarks and notes, etc.

The hypermedia model is very powerful and there have been interesting attempts to bring content creation for hypermedia in the reach of end-users. But the creation of good hypermedia requires a great effort in data acquisition[62] and is therefore difficult for unequipped user, as well as expensive. Most user have therefore remained hyper content consumers.

6.2.2.3 Visual Programming

Most programs nowadays provide users with a macro language that lets them specify repetitive tasks either by recording the user's actions or by programming by means of a macro programming language. These macro languages are primitive programming languages that are similar to languages used by professional programmers 20 years ago, e.g. BASIC, and lack most of the modern structuring and encapsulation techniques. This is definitely the wrong way, because these macros are based on primitive textual specifications.

A better way is to provide the user with a visual and interactive specification language, like visual query languages or specification environments for intelligent agents. These address only the information consumption part, but the same techniques may be used for content creation, e.g. in interactive Multimedia authoring systems.

There are also attempts for real visual programming environments that are not limited to a specific scope of applications. Visual programming languages (VPL) [Dan Kimura], [Dunne], [Glinert], [IEEE] provide a programmer with tools for laying out schemes of visually represented expressions. The importance lies in the fact that these graphics are executable. Especially useful are components for visual languages because they allow a programmer[63] to reuse solutions at a higher abstraction level. Visual programming languages are not only for end-users, but provide a way for faster and easier programming in general. Whether it is possible to replace text-based programming languages completely by graphic specifications is not yet clear. Theoretically, advanced programming paradigms can be matched to graphic expressions, e.g. object-oriented approaches[64], and in fact many of these modern systems make vivid use of graphical specifications.

[62] Data capture devices targeted at the public audience include digital cameras, microphones, video recorders, etc.

6.3 Categories

6.3.1 Hardware

The trend from multi-user systems to personal computers has long begun. It continues in moving the machines nearer to a specific user and it will soon be common for people to have multiple machines for themselves each suited for different tasks.

6.3.1.1 Light Pens & Touch Screens

Light pens and touch screens are adaptations of the conventional CRT displays to provide more direct manipulation. Light pens have been used for drawing and for digitizing data points. Touch screens are useful for presentations where users can only select certain topics. They require no secondary means of input, except for the user's finger. Both techniques are useless for handwritten input, though, because fingers are not precise enough and writing on a vertical surface is awkward, especially for small text.

6.3.1.2 Tablet & Screen

Digitizing tablets are more precise, and it is easier to write on a horizontal surface with a pen. Because the tablet and the screen are separated there is less interactivity, though. The user is forced to change focus between the two which is also not ideal for handwritten input.

6.3.1.3 NotePads

NotePads were the first systems that combined display and digitizer. The rest of the computer fits nicely in the same box. They are sized in imitation of letter paper, often with the same display resolution as desktop systems. NotePads are mobile platforms often tailored for specific applications.

6.3.1.4 Convertibles

Convertibles are combinations of the pen-only NotePads with the keyboard-only notebooks. The keyboard is either detachable or can be hidden under the display. Then, they look like NotePads. Convertibles are very popular for running

[63] Instead of "programmer" it would be more appropriate to speak of a "specifier".

[64] It is not difficult to imagine visualized objects, inheritance, and message passing. What is not so clear, however, is how all the dynamic aspects of a running program can be shown without loosing the overview even for large projects.

Microsoft's *Windows for Pen Computing* operating system, because it still relies on keyboard input (cf. section 7.1).

6.3.1.5 Personal Digital Assistants & Communicators

The Personal Digital Assistants are much smaller in size than NotePads and are intended for personal use at any place. They contain personal organizing tools, and combined with wireless communications capabilities, they build the platform for ubiquitous information access and exchange. When they are programmed for a specific application domain, PDAs can support users with their work, thus providing a complete productivity environment.

6.3.1.6 Wallboards

On the other extreme regarding their size are digital wallboards that are sensitive to pen input [Bruce, Elrod]. Some consist of large displays and some track pen movements on conventional whiteboards. In any case, they make up a platform for workgroups, either in a closed room or they allow for remote conferencing with people at distant places. When a display is used it can also serve to play back movies, pictures, and other computer data. There are also communication devices which allow simultaneous use of voice and data to allow for more intuitive cooperation.

The difference between wallboard conferences and video conferences is that the former concentrate on data and content and the latter concentrate on people. A combination or integration is possible.

6.3.2 Software

Software that can be used with pen-based systems is different from other software. Much of the difference lies in the user interface, but there are also structural differences due to the mobility and intuitiveness of these systems. In this section, the term "application" does not refer to application programs, but sets of tools, assistants, and agents for a specific task.

6.3.2.1 Horizontal Applications

Horizontal applications provide a base functionality that users can adapt for any problem. They build standard solutions for common needs. Text processing and spreadsheets are good examples of horizontal applications. The horizontal application *per se* in Pen Computing is the notebook. It allows the entry of any kind of data on an empty sheet of paper. There remains, however, the difficulty with the lack of context information in such an open environment.

6.3.2.2 Vertical Applications

Vertical applications are designed for clearly defined problems. They are very strict in their structure and are usually developed for a specific client or market. The advantage of vertical applications is that they are closer to the user's environment. The narrow focus helps make the applications easier to use and safer because there is more context information available. Vertical pen-based applications are already widely used – e.g. in medical environments (patient records, etc.) – and there will be many more to come.

6.3.2.3 Diagonal Applications

Diagonal applications emerge when the openness of horizontal applications is combined with tools to help a user customize the base functionality to build a vertical application. It is very difficult to give a user the power of flexibility without loosing ease of use. Drag-and-drop templates and components are examples for such power tools.

While horizontal applications are often implemented for small devices (PDAs) and vertical applications tend to be used on larger ones (NotePads), diagonal applications could integrate both.

6.4 Conclusions

The new paradigms for Pen Computing that have been discussed are partly emerging in desktop system, too. However, it is much easier and more natural to build new systems from scratch than to re-tailor existing systems. A seamless integration of all components is the benefit.

Where Pen Computing brings completely new fields for applications is in the way that everyone, everywhere, everytime computing is enabled, especially with *Personal* Digital Assistants. Personality builds a new element in computing that requires a lot of information about the user and the current environment. This information will help in providing superior user productivity and satisfaction.

7 Current Systems and Products

The following sections will provide details on the most important and/or promising systems that are currently available for Pen Computing. The goal is not so much to enlighten every detail of all existing platforms or to list all available systems, but to discuss the most interesting and innovative systems in order to allow for a comparison of the state-of-the-art of the existing systems with the previously outlined requirements.

The pen-based systems of Microsoft (*Windows for Pen Computing*), GO (*PenPoint*), Apple (*Newton*), General Magic (*Magic Cap*, *Telescript*), and some others will be discussed in the following sections.

7.1 Microsoft Windows for Pen Computing

7.1.1 Windows for Pen Computing

At a first look, Microsoft's Windows for Pen Computing [Microsoft], [Udell 1] is not different from the well-known Windows operating system. There is the same DOS layer underneath and the same user interface. Windows for Pen Computing consists of a few extensions that do not exist in the common Windows version: new mouse and display drivers, new control panel items, the PenPalette application in the startup folder, and new DLLs in the system folder.

The new mouse driver separates pen input for mouse emulation and actual pen strokes. The PenPalette application changes all Edit fields of open applications to HEdit fields when it is running. Edit fields accept typed text and HEdit fields accept handwritten input which is converted to text. Thus, existing applications instantly work with the pen without modification. On the other hand, this solution is not perfect, e.g. because the size of the Edit fields is not modified it is extremely difficult to write by hand in the very small fields intended for typed input.

In addition to the HEdit fields the Pen API also provides BEdit fields, i.e. fields for handwritten input where each character is written in a separate box. Applications that want to make use of BEdit boxes for more accurate recognition must be rewritten. Pen-specific applications[65] have more access to work with pen input. Such applications can also define dictionaries for limiting the acceptable input for a field (recognition context).

The handwriting recognition engine recognizes printed letters and is trainable. The Trainer application provides the interface. It allows the user to view and edit the recognizable characters together with the strokes that are accepted or to see a stroke and how it would be recognized. It is possible to define sequences of characters for a single shape, e.g. "er" or "th". The settings for a specific user can be saved in a file and loaded on a different machine. Thus, it is also possible that several people work on the same machine.

The recognizer is a Dynamic Link and Load (DLL) module and can therefore be easily exchanged as long as it provides the same interface.

The main reason for using Windows for Pen Computing is compatibility with the wide-spread interface on desktop systems. Common applications can be used and there are many developers and tools.

But Windows for Pen Computing hardly cares about pen-specific user interface issues. There is still the omnipresent cursor and the pen is mostly a replacement for the mouse. Even for the desktop version of Windows it is strange to see that the mouse plays only a marginal role. All commands can be issued from the keyboard, but not necessarily with the mouse. This situation is even more tragic when there is a pen but no keyboard available[66].

A newer version of Windows, currently known as Chicago, will incorporate the pen-specific aspects more smoothly. It is not clear, however, whether there will be significant improvements in the user interface.

An operating system for a hand-held device, called *WinPad*,[67] has been cancelled – *delayed* in Microsoft terms – because the Microsoft people were, of course, unable to squeeze Windows into a tiny elegant device.

65. Applications can register themselves to the system as pen applications using the `RegisterPenApp` function. They do not rely on PenPalette to be running for accepting pen input.

66. The author tested a demo application the window of which was the size of the screen in landscape orientation. This did not change when running Windows in portrait orientation (more useful for mobile devices) and therefore the Exit button was invisible. Without a keyboard it was impossible to quit the application without resetting the whole machine!

67. Only temporarily referred to as the *Newton killer.*

7.1.2 NCR NotePad

The NCR NotePad 3125 was among the first pen-based systems available. It was built with the Intel 80386 processor (cf. section 3.3.2.1) and featured a 640 x 480 pixels large digitizing display, a hard disk, and many slots for external devices. The later model (NotePad 3130) added backlighting to the display. Still, the machine is quite heavy and the processor is not ideal for handwriting recognition. The NotePad runs either Microsoft Windows for Pen Computing or GO PenPoint.

The NotePad is still legendary for its early success in the Pen Computing world.

7.2 IBM's Pen Platforms

IBM is involved in several pen-oriented projects and is pushing this new technology in its product line. On the operating system side, there are two options to choose from: *PenDOS* and *OS/2 for Pen*, both of which are extensions to existing operating systems – an approach much like that of Windows for Pen Computing.

7.2.1 PenDOS

PenDOS was developed by CIC, who built it around their well-known (letter-by-letter) HWX engine. There is a collection of development tools which make use of the API. There is also software for signature verification, as well as a trainer program for adapting the HWX engine to a specific user.

The greatest advantage of using pens in a DOS environment are to be seen in the low hardware resource requirements – making it fast and efficient – and the large number of installed systems.

7.2.2 OS/2 for Pen

IBM's Natural Computing Technology department focuses on several kinds of human-centered technology, especially speech and handwriting recognition. The HWX engine of OS/2 also processes only letter-by-letter input, though. These technologies will be provided as (OS/2 Workplace-) personality-independent services that can be combined with user-interface agents, tele-conferencing tools, etc. IBM's hardware platform of choice is built around the PowerPC. It provides the power that is required for such computation intensive tasks. The advantages of OS/2 are especially evident in its communications and multitasking capabilities.

Although OS/2 usually runs Windows applications, it is not possible to use applications that were specifically written for Windows for Pen Computing on OS/2.

7.2.3 IBM ThinkPads

IBM entered the Pen Computing market only much later than NCR, but with a more sophisticated model, of course. The ThinkPad 710T [IBM 1] was among the first to feature the Intel 80486 processor, which makes it a lot faster. The display of the ThinkPad is much superior to NCR's NotePad because its surface is rough and therefore feels more like real paper, which is not as slippery as glass, but resistant. The rest of the design is similar to the NotePad's and runs either OS/2 for Pen, Microsoft Windows for Pen Computing, or GO PenPoint.

The ThinkPad 750P [IBM 2] is a convertible that looks like the 710T when folded. The display can be lifted to uncover a keyboard. The system then looks like a regular notebook. The ThinkPad 730T is also a convertible which features a color display in addition.

7.3 GO PenPoint

7.3.1 PenPoint

PenPoint by GO Corp. [GO], [Carr], [Carr, Shafer], [Clarkson], [Novobilski] was the first operating system explicitly developed for Pen Computing. It broke ground in several areas, especially in establishing the awareness of Pen Computing as a serious human-machine interface technology.

7.3.1.1 User Interface

The PenPoint user interface is graphically as well as functionally based on the generally known paper notebooks. Of course, the Notebook User Interface (NUI) does more than a paper notebook, but the interaction schemes are very similar. The user is first presented a page with a table of contents which shows all the documents the notebook contains. The documents can be grouped in chapters (folders) and viewed by tapping their icon, title, or page number and there are tabs on the side of the page which allow for direct access to a chapter or document. Unlike the paper model, the digital notebook supports scrolling and menus at the top of the page.

At the bottom of the screen there is an area called the "bookshelf" where the notebooks can be accessed by tapping their icons. There are also some system functions put on the bookshelf.

Documents are created by selecting a document type from the "Create" menu. Hence, applications are not visible, but activated whenever a document contains data of the respective type. Documents of one type may contain documents of another type. Whenever a document – or a document inside another document – is activated the corresponding menus and toolbars appear. This recursive embedding of documents is called the Embedded Document Architecture (EDA).

There are also links (reference buttons) that can be created to access any part of a document by a tap of the pen.

Handwriting recognition in PenPoint works character-per-character. In order to edit a word the user is required to draw an edit gesture (a circle) over the respective word. A window opens where each letter is printed inside a box. The individual letters can be overwritten here. This method is of course used to make recognition easier, because the positioning of the letters is exactly known, but for the user this approach is somewhat unintuitive. The text that can be edited is different in look and size from the text in the document and rearranging words is difficult. The HWX engine is exchangeable to allow other developers integrate their recognizers seamlessly.

A whole set of gestures is built into the PenPoint interface and more can be added to suit special applications. The 11 core gestures allow for inserting, deleting, and editing words and letters, and scrolling.

Among the system functions that are accessible from the bookshelf are the In and Out boxes. There are the containers for messages from and to the outside world. Because the mobile devices are not necessarily permanently connected to a communication network the messages – faxes, e-mail, paper prints, pager messages, etc. – are kept in the Out box until a connection is established[68]. The In box contains messages the user has received but not processed yet.

7.3.1.2 Operating System

Apart from its dedication to pens, PenPoint is a powerful operating system as such. It features a 32-bit architecture with preemptive multi-tasking and virtual memory support.

PenPoint's nice user interface is thoughtfully mapped to an object-oriented class framework which implements all the general interface elements. The object-orientedness of PenPoint's framework is implemented by the ClassManager. The ClassManager is a module written in ANSI C and it provides for all the object-oriented techniques like inheritance and message-passing. Due to its standard base, PenPoint is very portable. On the other hand, for the programmer using the ClassManager routines is uncomfortable because it does not offer a proper syntax[69] and keeps all of C's ugly constructs adding some more. A pre-compiler would have changed this situation, but portability obviously was the most desired feature.

Permanent data storage in PenPoint is done by writing MS-DOS compatible files which is again not very elegant, but a great plus for compatibility must be attributed. The use of conventional files is nonetheless not suited for object-oriented programming.

The advantage of the concerns for compatibility and portability was showed when PenPoint was ported from the Intel 80386/486 architecture – which it initially was developed for – to AT&T's Hobbit architecture in a short time.

7.3.2 EO Personal Communicators

GO's PenPoint was originally written for Intel-based machines, but the PenPoint machine *per se* is the EO. The EO [EO] is a tablet with ears and incorporates AT&T's Hobbit processor (cf. section 3.3.2.2) for high performance. The operating system is burnt into ROM and allows for short start-up times. The main focus of the EO is on communications: it comes with a built-in wireless phone and fax capability.

68. Connections can be established and quit while the system is running and data is received automatically – this is very useful for mobile and wireless connections.

69. E.g. a true OO language offers a syntactic structure for message-passing like
`myObject:doWhatIWant(613);`
which must be written in PenPoint's ClassManager like this
`ObjCall(doWhatIWant,myObject,613);/*`
ObjSend in some cases */
Class definitions and object instantiations are even worse.

7.3.3 Conclusions

PenPoint is a powerful general purpose operating system with special care for mobile use. The problem with PenPoint is that it is neither as compatible as Windows for Pen Computing nor as revolutionary as the Newton technology. The same is true for the EO: too small for a desktop replacement and too large for taking along. There are still several PenPoint developers, but they are facing a difficult situation.

In the beginning, GO relied only on the operating system itself and did not develop hardware. When the EO established as *the* PenPoint machine and other platforms tended to favor Windows, GO and EO got closer in their growing isolation. EO was also the only company to use AT&T's Hobbit chip. The conclusion was that GO took over EO and AT&T took over GO. Unfortunately, AT&T closed EO as a result of the poor sales of their communicators by the end of July 1994. AT&T still has all the potential to promote PenPoint on a large scale, but they are involved in other important activities, too. AT&T also took over NCR and, more important, is a major partner of General Magic. AT&T favors General Magic over its contenders because of their advanced Telescript networking concepts. The situation for PenPoint is that of an extinct animal. All the same, AT&T is strongly committed to the personal wireless communication market.

7.4 Apple Newton

7.4.1 Newton Technology

Apple's Newton technology [Apple PIE], [Betz] is a hardware-independent concept for a pen-based environment. The technology is licensed to other companies that want to develop machines based on this environment. This open architecture guarantees for the development of many compatible devices where each company can integrate its experiences. The high expectations for service integration make this approach a necessity. Newton technology is intended to work for devices of any kind and size (from wrist watch to wallboards).

With the Newton technology, Apple is treading new ground and provides an environment free from compatibility concerns with its Macintosh or other computers.

7.4.1.1 User Interface

The Newton user interface basically consists of a sheet of paper. Of course, the number of sheets is not limited and the sheets can be organized in folders by assigning a sheet to a folder in the filing button (a pop-up menu in the upper right corner). A new sheet is created when a horizontal line (gesture) is drawn over a visible sheet. The info button in the upper left corner of a sheet when tapped displays the date and time when the sheet was created and the storage space it takes up. Next to the filing button the routing button is placed (shown with an envelope icon) with a pop-up menu that allows the corresponding sheet's contents to be printed, faxed, beamed (via an infra-red connection), or e-mailed (via NewtonMail). Several print and fax templates are predefined. When an external connection is not available the data are placed in the Out box.

The contents of a sheet can be easily manipulated by simply writing words, graphics, and other drawings. Depending on the recognizers[70] currently activated the drawn lines are interpreted. There is no specific mode for entering text[71] and it is also possible to apply deferred handwriting recognition.

Text, graphics, and drawings can be selected by holding the pen tapped on the display until the selection mode is entered and then circling the objects or drawing over the text. Deleting an object works by scrubbing it out[72].

The Newton user interface is designed with great emphasis on handwriting recognition, which is best illustrated by the Intelligent Assistant (IA). The IA is accessed by selecting a phrase and tapping the *Assist* ( ) button. The Assistant then searches the phrase for a key word that it can interpret as a command and for associated parameters. Some Assistant key words are built in[73], but they can be programmed at free will[74] to perform complex operations. and toolishness are more appropriate terms for Newton's functionality than applications, because Newton concentrates on data.

Some system functionality is accessible through the Extras drawer and an address list and a diary with a to-do list are also built in.

7.4.1.2 Operating System

The Newton's operating system, called "Newton Intelligence", features 32-bit addressing and multi-tasking, but most important is that it is object-oriented from bottom up. The basic means for this approach is the dynamic object-oriented language *NewtonScript* [Evins] – a descendent of Self[75] [Ungar, Smith] with Pascal-like syntax. Self is different from Smalltalk and object-oriented LISP dialects because there are objects but no classes. Self eliminates the static nature of classes by inheriting from prototypes. Prototypes are more dynamic than classes because they can be altered at run-time together with inheritance. The prototypes are objects themselves and ordinary objects are simply copies of prototypes with a pointer to their origin[76]. Objects are implemented by frames (cf. later).

70. In the lower left corner of the display a series of icons represent the various recognizers that can be activated, currently e.g. for cursive handwriting, letter-by-letter handwriting, and graphic shapes like rectangles, lines, and circles. It is preferable to activate only one recognizer at a time for better recognition results. If only e.g. the recognizer for cursive handwriting is activated all lines that are drawn in the input area are converted to words that are stored in the Newton's dictionary.

71. Except for the choice of recognizing words based on a customizable dictionary or on a letter by letter basis.

72. Deletion is animated by a vanishing cloud!

73. For example, the phrase "David lunch tomorrow" searches the name file for a person called David and inserts a meeting with him at noon the following day in the diary (not without asking the user for permission, of course).

74. Unfortunately, this is only possible for developers using the Newton Toolkit (NTK) and not for end-users.

75. Self was developed at Stanford University and Sun Microsystems Laboratories. There have also been great influences from Dylan, an object-oriented dynamic language based on CLOS, but more dynamic. Dylan is still being developed at Apple's Eastern Research and Technology group in Cambridge.

76. The difficulty with classes that are instances of metaclasses, etc., is therefore avoided.

Despite its name, NewtonScript [McKeehan, Rhodes] is a full-featured programming language. The attribution "dynamic" stems from the fact that NewtonScript is only marginally compiled and to a great extent interpreted at run time[77]. NewtonScript's primary data type is the frame. A frame is an unordered collection of slots. The slots are accessed by their name any may contain data of any type. Frames can be changed at run time, e.g. assigning a value to a slot changes its current value and creates it if it does not exist yet. The slot values can be simple integers or strings[78], or they can be frames or NewtonScript function[79]. There is no strong typing, even data types are dynamic and determined only by a slot's value[80].

Programming for a Newton-based device is done using the Newton Toolkit (NTK) on a Macintosh or Windows-based computer. The NTK shows a representation of a Newton display and a palette of user interface elements that can be dragged to the display. Code is edited in a Smalltalk-like browser. All visible (and some invisible) user interface elements are contained in views. Views are frames with special properties that are inherited from the Newton's prototype framework. Inheritance is implemented in NewtonScript by slots that contain pointers to other frames (= objects). There are two kinds of inheritance: parent inheritance and proto inheritance.

- Parent inheritance is used to represent the hierarchical structure of the views and implemented by the _parent slot of a view frame, e.g. a RadioButtonCluster containing several RadioButtons.
- Proto inheritance is implemented by the _proto slot. This inheritance mechanism gives access to the Newton's prototype framework where each user interface element is implemented by inheriting from other elements.

Messages that are sent from one object to another are implemented in the target object itself by a slot with the message name (= method)[81] that contains a NewtonScript function or inherited. Inherited methods are executed when the interpreter cannot find a slot with the message's name in the frame and searches first all inheritance links following the proto inheritance and – if not found – all proto inheritance links of the original frame's parent, etc. The proto framework is typically in ROM, but all other inheritances can be changed at run time.

Data of NewtonScript programs is also stored in frames. If the data are to be stored permanently, a permanent object store (POS) can be created by collecting sets of frames into soups. Soups are independent from a specific program and allow for easy sharing of data. The soups provide for automatic compression and indexing and are finally stored physically in stores. Data inside a soup is retrieved by a query function.

Newton's handwriting recognizer was developed by Paragraph, a company from Moscow that was acquired by Apple after the iron curtain fell. Paragraph developed handwriting recognition algorithms for the Soviet postal services for reading handwritten addresses. The engine is able to recognize cursive writing by looking up words in a dictionary. This limits the recognizable words, but it is a good way to make the very flexible input method possible. A letter-by-letter recognizer has been added for writing rarely used words, such as names.

7.4.2 Newton MessagePads

The Newton MessagePad from Apple is the first implementation of a device based on the Newton technology. It features a 7.5 x 11.5 cm display (240 x 320 pixels) and weighs about 500 g. The machine is running on ARM's RISC processor 610 (cf. section 3.3.2.3, [Redfern]), which provides the computing power necessary for HWX and supports the NewtonScript language with advanced memory control for garbage collection. One PCMCIA slot (Version 2.0, Type II) is included for additional software, memory, or pager and modem cards. The operating system is stored in a 4 MB ROM with additional 640 KB RAM. The latest model (MessagePad 120) comes with 2 MB of RAM, a new case with a removable lid and improved screen visibility, as well as longer battery life.

The PCMCIA slot is also powered at a higher voltage to allow for more complex communications cards. One such card offers GSM connectivity for the MessagePad in Europe when it is connected to a Nokia GSM phone. There are also pagers and data/fax modems available on PCMCIA cards.

7.4.3 Motorola Marco

Marco is the first fully wireless Newton device. It is built on the same hardware architecture as Apple's MessagePads, but includes in its bigger case a wireless modem that connects to ARDIS and RadioMail. The support of RAM Mobile Data and CDPD is planned to be added. Marco allows faxes to be sent and email to be sent and received wirelessly.

The inclusion of true wireless data communications opens opportunities for a new class of applications.

7.4.4 Apple Mobile Messaging System

The Apple Mobile Messaging System is a multi-platform approach to integrate wireless communications services. It supports Newton devices, Apple PowerBooks, and notebooks and -pads running Microsoft Windows. The Apple Mobile Messaging System combines a pager card with appropriate software that allows for sending and receiving emails – personal or from information services – via PageNet's paging network. A wired connection is needed for sending mails because pagers can only receive data wirelessly. The idea behind the system is to make it easy to use the currently bewildering variety of devices and protocols that are available.

77. The interpreter resides in a ROM.
78. Strings are coded in Unicode, although it is not comfortable specifying them, e.g.
`myName:="Andr\u00E9";`
and only a subset is supported for input, namely the Apple Extended ASCII character set, further restricted in American version which do not support European characters like German umlauts or French accents for HWX (they are only accessible through the use of the on-screen keyboard).
79. Like in C, every NewtonScript function returns a value.
80. Local variables are typed by assignment, too.
81. There is no distinction between a slot containing values (a.k.a. instance variable) and one that contains a function (a.k.a. method). Slot contents can be changed at run-time.

7.4.5 Conclusions

Apple's Newton technology is a revolutionary and convincing approach to Pen Computing. It offers most of the discussed features, although the currently available MessagePads are not perfect. Future models will probably support more communications features, such as Telescript, and add much functionality which is easy due to the clever user interface and system structure. A larger Newton model, known as the Slate, with a VGA-sized display will soon be available by Panasonic. This demonstrates the flexibility of the technology and offer additional opportunities in vertical markets.

7.5 General Magic

7.5.1 General Magic

General Magic's staff [General Magic], [Halfhill, Reinhardt] is a collection of some of the most marvellous and creative minds of computer science. Andy Hertzfeld and Bill Atkinson formed the driving force behind the Macintosh and together with Marc Porat try to go much farther on that way now. Unlike the Newton, General Magic was spun off from Apple because the scope of the intentions was too wide to be covered by a single company. So they joined with AT&T, Matsushita, Motorola, Philips, and Sony to form an independent development facility. General Magic's licensees will bring products to market that make use of its technologies. This way, they need not care about production concerns and can guarantee their developments to be widely available.

General Magic sees a person's activities in three overlapping aspects.

- Remember
  The little personal things everyone has to do constantly from day to day.
- Know
  The structure of things in the world and general information regarding many individuals.
- Communicate
  The links and relationships between individuals.

Around these activities General Magic's technologies will circle to enhance people's potentials.

7.5.1.1 Magic Cap

Magic Cap (Communicating Applications Platform) [Knaster] is an interface and development platform for communications-oriented environments. Magic Cap starts from the Mac's desktop metaphor and provides the user with an interactive environment not only of a desk, but of an office, house, and streets, too (all in 3D)[82]. Buildings in the streets represent information providers. Depending on the services accessible to user, they can design their own office and streets.

Magic Cap does not include tools for creating documents, except for memos, postcards, etc., and lacks handwriting recognition[83]. Instead, it is easy to communicate with other people by electronic mail and to find information on worldwide data networks. The second technology from General Magic, Telescript, makes this possible.

Magic Cap can be used in various environments, from PDAs to telephone and fax machines or for Interactive TV[84]. It will also run on top of existing desktop platforms such as the Macintosh or Microsoft Windows.

On the developer side, Magic Cap provides an extensive object-oriented framework built on the Macintosh Programmer's Workshop (MPW) in C++. Unfortunately, developer support from General Magic is very limited. There are only a handful of developers for Magic Cap at the moment.

7.5.1.2 Telescript

The current situation with electronic information access is so confusing for programmers as well as for users. An almost unlimited number of services and protocols exist and most of them are only accessible by a variety of primitive text-oriented interfaces.

Telescript [General Magic], [Gore], [Wayner 4] attempts to solve this incoherence by providing a single programming interface similar to and inspired by PostScript. In fact, Telescript shall do for communications what PostScript did for printing, namely provide an easy yet powerful and unified specification language. Telescript is a dynamic[85] object-oriented language and requires a run-time interpreter – the Telescript engine – that is independent from the current environment. Hence, Telescript programs will be able to run on any platform. Telescript's platform is in fact not a single computer, but the network as a whole.

The three constituent parts of a Telescript network are agents, places, and go. Agents go from place to place where they meet other agents and perform operations that were specified before[86]. Places may be grouped into electronic marketplaces. There, agents can go shopping or retrieve useful information. The currently bothering issues in communication, such as naming, addressing, routing, and security are hidden. Another important difference is in the establishment of a connection. Usually, a connection has to be kept running during the time of information retrieval, whereas with Telescript only a short time connection is required to transfer an agent which then runs on the server device[87]. This works, because Telescript does not transmit raw data (texts in most current systems), but programs.

82. *Desk, Hallway,* and *Downtown* in Magic Cap's terms.
83. The only way to input text is by means of an on-screen keyboard, except for digital ink.

84. Rumors about MagicTV already appeared.
85. This refers to the ability to define and redefine classes at runtime. In addition, Telescript programs are persistent, i.e. the do not lose state and data when a device is switched off.
86. "Go" and "Meet" are commands of the Telescript language.
87. The Remote Procedure Call (RPC) concept requires client and server to constantly exchange information with each other. General Magic's Remote Programming concept requires a connection only for the transfer of agents. The agents work on the server machine to interact with the services that are available there. Similarly, servers may send agents to clients. This way, the network traffic is minimized, which is especially important for wireless connections.

When Telescript will be widely supported (AT&T PersonaLink Services are the first Telescript networking services available), there will also be a chance for a market for agents, i.e. agents that have been programmed for specific tasks can be sold like today's application software.

7.5.2 Motorola Envoy

The Motorola Envoy is the first Personal Intelligent Communicator (PIC) that uses Magic Cap and Telescript. It is built on Motorola's "Dragon" processor[88] (cf. section 3.3.2.7) and features a 480 x 320 pixels large digitizer/display.

The device includes a 9600/2400 bit/s fax modem, a 19.2 kbit/s wireless data modem providing access to ARDIS, two PCMCIA type II slots, an infra-red transceiver, a microphone and speaker[89], and various software, e.g. eShop that is part of AT&T's PersonaLink Services, RadioMail, etc.

7.5.3 Sony Magic Link

Sony's Magic Link is almost identical to the Envoy, except for its lack of two-way wireless communications (ARDIS) – instead, it conforms to Sony's IR interface – a different case, and lower price.

7.5.4 Conclusions

Magic Cap is a nice interface for dealing with various means of communication. On the other hand, it seems to be somewhat limited in dealing with larger documents.

Telescript, if things happen as they should, will be the major revolution in information retrieval of the next decade. It is integration and platform independence are the keys to ubiquitous worldwide communications.

One question with regard to Apple's dedication towards Magic Cap is unanswered: how will it compete with the Newton which provides a more general approach to the user interface? Telescript will be supported in future Newton and Macintosh models without any doubt.

AT&T's conflict with GO/EO, NCR, and General Magic was already mentioned. With the PersonaLink Services AT&T already supports Telescript.

7.6 Amstrad PenPad

After the expectations went high for something called a PDA, Amstrad were the first to actually have one ready for sale. Still, the Amstrad PenPad is the cheapest and lightest of all. This can, of course, be explained by the simplicity of its design.

The heart of the PenPad [Amsoft], [Pountain] is made up of three Z80-compatible processors. These are 8-bit architectures. They run in parallel and are functionally separated in power management, character recognition, and data processing.

[88] Magic Cap will probably soon be ported to the PowerPC, too.
[89] For recording and playing back spoken messages without speech recognition.

Text is entered in boxes, letter-by-letter, and the handwriting recognition engine is trainable. The PenPad uses a proprietary operating system and software (diary, to do list, calculator, etc.) is hard-wired, i.e. printed icons are located around the border of the display. The PenPad features system messages in five languages (note the European approach), a slot for PCMCIA cards of type I, but lacks communications, except for a serial connector.

The PenPad can be programmed in Assembler, C, or Pascal using Amstrad's Application Program Interface (API)[90] and an MS DOS-based Z80 compiler. The developed software can be accessed from the PenPad when it is located on a PCMCIA card.

Amstrad announced a newer model for mid-1995: the PIC700 *InfoPad*. It features a larger display, a Hitachi RISC processor, and support for wireless communications (paging and GSM). The focus has shifted from a simple notebook to mobile communications.

7.7 GeoWorks GEOS

7.7.1 GEOS

The GEOS operating system was first written for the Commodore 64 and it was impressively efficient even then. This efficiency was also a major benefit when it was later ported to DOS-compatible machines where it runs on top of DOS much like Microsoft Windows but with much better performance[91]. GEOS' clever design is now available for Pen Computing.

The user interface is an adaptation of the Motif look with handwriting recognition for hand-printed characters.

The operating system is built on top of DOS in a six-layer model (hardware, DOS/BIOS, drivers, kernel, dynamic libraries, applications). The file model is also taken from DOS, which is certainly not the file system of choice for an intuitively usable environment. The layer model of GEOS allows applications written for the desktop version to run on the mobile version without recompilation. On the other hand, the software is written in Assembler because there is no developer environment available. Third party software is therefore hardly to be expected soon.

The only choice for higher-level programming on the Zoomer is available in the form of the Interpreted Zoomer Language (IZL), a BASIC-like interpeter, which is hardly powerful enough for serious application development, though.

7.7.2 Casio/Tandy Zoomer

The Zoomer is a handheld pen device built on an Intel 8088-compatible processor made by Casio. GEOS is the ideal operating system for this low tech 7.5 MHz, 16-bit processor and it

[90] The API is built into firmware in the PenPad and provides about 120 routines of all kinds. It hides the multi-processor architecture completely.
[91] GEOS can be used on lowest performing DOS machines, features multi-tasking, long file names, scalable fonts, and many other things with very low memory requirements. It does not provide many of Window's features, though.

is due to its efficiency that battery life reaches 100 hours. On the other hand, the processing power is also very low, a crucial problem especially for handwriting recognition and instant user feedback. The processor runs only in real mode, thus its impossible to shield off handwriting recognition or operating system actions from user processes as would be possible when using a protected mode.

There are lots of built-in applications in the Zoomer: an electronic mail front-end for America Online, a language translator for 26 languages, a thesaurus and spell-checker, a personal finance management program, and many other utilities.

Although the Zoomer is physically similar to the Newton MessagePad (in weight and size), it is more entertainment-oriented (two game buttons make it look like a Nintendo GameBoy) and the design emphasizes more on efficient use of low-tech components than on ground-breaking technology.

7.7.3 Sharp PT 9000

Sharp, the producer of Apple's Newton MessagePad and the almost identical Sharp Newton ExpertPad, also sells a GEOS-based machine, the Sharp PT 9000 [Perratore]. The PT 9000 is very similar to the Zoomer, only the larger display (640 x 400 pixels) makes a difference. The PT 9000 is less portable than the Zoomer due to its increased size but the enlarged working area offers possibilities for other applications.

7.8 Other systems

7.8.1 Xerox Tab

The Xerox Tab [Weiser] is a unique non-commercial product that was developed for use within the Xerox PARC itself. The Tab fits in the palm of a hand and features a small display and three buttons that are operated with the fingers while the Tab is held in the hand. The small size of the display led to the development of the unistrokes that are used for textual input. Within PARC, the Tabs of all the researchers are connected wirelessly. This allows for location and environment information to be shared. The Tabs support email and are used, for example, for controlling the air conditioning, lighting, and other environment parameters in a room.

7.8.2 IBM TouchMobile

The IBM TouchMobile [IBM 3] system is a rugged handheld device designed for use in corporate environments with mobile employees. The device runs DOS on an Intel 8088 processor (4.77 or 8 MHz) and features an infra-red transceiver for connecting to a docking station array for 16 mobile TouchMobile units. A server with corresponding software controls the data sharing. There is also a docking station for cars. The TouchMobile does not provide handwriting recognition, it is made for vertical solutions and has a bar-code reader built-in.

7.8.3 IBM/BellSouth Simon

Simon [Bell South] looks like an ordinary mobile cellular phone. That is what it is, but there is more. There is a pen-sensitive display between the speaker and the microphone which can be used to retrieve stored telephone numbers, draw notes that can be faxed. Simon is also a calendar, an address book, a pager, a calculator, and provides for wireless electronic mail – a mighty phone, indeed. Pen input is stored as digital ink and there is an on-screen keyboard for text entry. Simon was developed by IBM and is initially marketed by BellSouth Cellular Corp.

7.8.4 CIC MacHandwriter, Handwriter for Windows

The Handwriter is a very flat digitizing tablet with a cordless pen for use with either a Macintosh or Windows desktop computer. The accompanying software emulates the pen as a mouse and supports handwriting recognition. Existing applications profit from pen-input this way but the separation of tablet and display remains a problem.

7.8.5 Wallboards

There are a variety of different technologies that are used for various kinds of electronic pen-based wallboards. Some of them are presented in the following paragraphs.

7.8.5.1 FutureLabs TALKShow

TALKShow is Windows software for working on documents with other people over networks and telephone lines. People sitting in front of their PCs at distant locations can work together as if they were in the same room.

7.8.5.2 XEROX Liveboard

A large pen-sensitive display hides a computer running Windows [Xerox]. The applications can be controlled by pens while the interaction is transmitted to remote installations. Animation and sound can also be used to work concurrently over large distances.

7.8.5.3 Smart Writeboard/2000

Smart's WriteBoard is a pressure sensitive whiteboard (a large digitizing "tablet") with standard marker pens to write on. The drawing on the board is simultaneously redrawn on the screen of an attached Macintosh or Windows desktop computer from where it can be stored or transferred to remote locations.

The Smart 2000 system consists of a computer with a large pressure-sensitive display. It runs Windows and can therefore be used to manipulate data using standard software. Again, this system allows for remote conferencing.

7.8.5.4 Microfield Graphics SoftBoard

The SoftBoard is a conventional porcelain whiteboard with marker pens to write on. It is combined with a scanning system consisting of two lasers in the upper left and right corners of the board. The pens and the eraser have a reflective surface[92] so that the laser beams are reflected to determine their position. The data points (up to 80 per second) are transferred to a Macintosh or Windows desktop computer where the drawing is recreated. The resulting drawing can be stored for later use and transmitted via telephone lines to serve as a conferencing system.

92. Each has a differently patterned surface in order to be distinguishable by the lasers. This allows for "colors" to be recognized.

7.8.6 IDEATOR

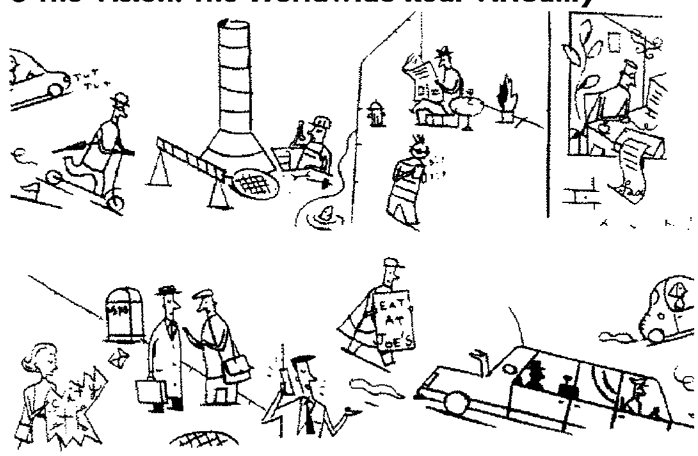
The IDEATOR is a concept for a sketching tool for industrial designers developed at the Delft University of Technology [Kolli, Stuyver, Hennessey]. This device is the result of a study of designer's working habits and features some interesting details. The entire sketching unit consist of three parts:

- Pad
  A digitizing display of size A3, commonly used in landscape orientation. The tablet can be rotated to fit any desired orientation and moved around on the desktop. There are some buttons on the Pad for special setup options and a microphone and speakers, e.g. for voice annotation.
- Pens
  A set of four different pens is used for different tasks. The pens are physically different in order to allow for real feeling of the pens. The functions are those of a pencil, a marker, a fine-liner, and an annotation pen, each with the respective color and thickness. The designer draws with the tip of the pens. The pens can also be turned around and the designer can then select portions of the image with the pen's end.
- Paw
  The second hand is less active than the first and operates a detached device called the Paw. This is a locally fixed "box" with buttons that allow for browsing, cutting and pasting, and control of connected VCRs, computers, etc.

The IDEATOR is specially designed to support designers in the very first phase of generating ideas. Hence, it must impose very few restrictions on how the designer works. In addition to the real pen-and-paper environment the IDEATOR provides for moving and resizing pictures – using the pen's end – as well as for animation – using the paw. The animation feature can be used in different ways: animation of sketches (redraw in timely order of generation), animation of parts of sketches (to show how a designed object behaves), and video playback (to combine sketches with real-world images). All interaction techniques are strictly image-based, i.e. there is no recognition capability. This is because the sketches that designers produce are not formal.

The IDEATOR demonstrates how closely a pen-based device can match its environment when the working habits are thoroughly studied. The narrower the application field is defined the better can the device be adapted to the users.

8 The Vision: The Worldwide Real Virtuality



The key effect of all the advances in different technological fields will come into play when they are all together seamlessly integrated in a coherent and transparent way. Then, the user will not need to know any details of different technologies, but will understand them and make use of them immediately. First, a user – seen as a *consumer* – needs freedom of choice. Second, as a *producer*, a user needs freedom of creation. Both kinds of freedom do not exist currently – not necessarily because it is impossible to consume and produce information, but because it is much too difficult as well as cost and time expensive even for skilled people. The necessary shift of user[93] behavior towards free information exchange takes place on three different levels.

- Individuals
  There is a change in communication between individuals. Nowadays, the telephone is the only reliable means of communication. Telephones are accessible from all around the world. On the other hand, telephones cover only a small part of the interaction, namely speech, and they require the two actors to be connected simultaneously. Combined speech, facsimile, and data connections start being available.
- Workgroups
  There are hardly any companies whose workers work isolated. Typically, most employees have several meetings a week, sometimes far away from their office. Travelling to numerous places takes up much time and energy. And it is not always easy for people to present their work and ideas convincingly at remote locations where they lack their accustomed environment. Of course, personal contact is necessary and useful, but no products are created or sold with lunch meetings alone. Workgroups that can truly collaborate globally have the advantage of concentrating on their work. Video conferencing is only part of this, more elaborate techniques allow people to share documents and work simultaneously on them at remote locations. This way of work frees employees from being at a certain location and allows specialists to assist in multiple projects without mov-

93. The user is considered here always in the twin role of a consumer and producer. With this extension, the otherwise old-fashioned word "user" is still valid.

ing around.

- Mass Media, Information Providers
Mass media are still considered only as a source of information, where the freedom of choice is restricted to switching TV channels and subscribing to newspapers and journals. The latter have the advantage of accessibility at any point of time.
Anyway, the information anyone can access is always in a final form of presentation. The only choice left is whether to read an article or not. The presentation should be more flexible, providing basic facts and means for putting them together. This may be very dangerous – for political news, for example – because journalists are responsible for the accuracy of information. Bits of information cannot be put into a context deliberately. People also want points of view and interpretations instead of pure numbers. On the other hand, much of the information people need is not so biased. The problem there is mostly one of putting things together. This can be supported by tools that help in locating information on networks, data bases, etc. Herewith, both the consumer and producer aspects are covered and something like an *Information Warehouse* will come into existence.

There are three core concepts which are required for the previously outlined shift in dealing with information: Multimedia, interactivity, and mobile communications.

The integration of all media renders the freedom of choice possible. Interactivity is required for the freedom of creation. Only together, though – and paired with unlimited information access by means of mobile communications – they will create a virtual information space which will be part of the user's real world environment. Hence, it can be called a *Real Virtuality*, opposed to Virtual Reality. While the virtual space, or *Information Space*, created by Pen Computing and mobile communications exists in the real world, the Cyberspace of Virtual Reality forces users to enter an artificial environment physically. In the same way as a Cyberspace extends and replaces physical parts – e.g. by implanting additional functions into the brain – to create a physical tele presence (for remote surgery, etc.) the Information Space generates a tele presence of the mind[94] [Minsky]. The Information Space is a silent revolution[95] compared to Cyberspace – it is unobtrusive technology that fits into the human behavior. Besides that, the Cyberspace does not work in all possible situations – e.g. climbing virtual stairs is physically impossible. On the other hand, there are no restrictions on the state of the mind.

The benefit of mobile computing is twofold: users need not go where they do not want to go and they can go where they want to. With a small ultra-personal device, like a PDA, a user will be able to connect to all kinds of services at any location. For services that require different hardware – e.g. large or touch-sensitive displays – the user will be able to connect to such devices that are accessible at many places – like telephone cells are today – to make personal settings accessible to different platforms. [Weiser] refers to this as *Ubiquitous Computing*.

Pen input and handwriting recognition build the most flexible, precise, as well as the most generally applicable interaction platform. The sum of all previously described concepts and techniques is referred to as *Interactive Paper* by the author – i.e. a platform which is as easy to use as paper, due to the implementation of the outlined interaction techniques, and allows a user to deal with data of any kind, no matter where in the world they are stored.

A different aspect of increasing importance in the Information Space is entertainment. This is because there is a general desire for amusement that increases as people content with short-term pleasures [Minsky in Riecken]. Today, entertainment is only passively enjoyed. Movies, television, etc. are offered in a take-it-or-leave-it manner. *Interactive TV*, *Video On Demand*, and other buzzwords introduce a new perception of entertainment as a source and field for interaction. Television on Interactive Paper, for example can get away from sequential storytelling. Along with a movie, there may be various plots or background information included that can be explored[96]. The already difficult distinction between information and entertainment will become even more difficult. It remains to be seen whether people will enjoy the interactivity of entertainment, especially when extends beyond video games to allow for creativity. Maybe, people will merely enjoy more sophisticated presentations still passively. It is not easy to get users to act. When content creation becomes so easy that people's creativity is stimulated, there will hopefully no longer be a distinction between users and creators.

It is very important to mention that intuitiveness and ease of use do not mean that anybody can do everything without training. Rather, education is definitely required, for using the tools and especially for dealing with information. Intuitiveness means that the tools are closely matched to the real-world tasks and that the human capabilities are not restricted by technological barriers. The ways of interaction are seamlessly adapted to humans and problems.

Especially in education – one of the most often cited fields for the application of Multimedia technology – the importance of qualified training cannot be underestimated. Without training, people will not improve their creative expressiveness but degrade to a level of Artificial Primitiveness.

If people get the tools and the education, so they can work with pens[97] in the way they are used to, a return of cultural awareness and concentration on contents may result. This is what the Vision in the subtitle of this paper refers to.

94. Because the new devices have access much more information on their user they provide a virtual personality including the user's interests and knowledge.

95. "Newton is low-impact technology, something that helps you keep moving rather than tying you down", Steve Capps in his commentary to "Newton's Law" [Gore, Ratcliffe].

96. For example, *The Complete Maus – A Survivor's Tale* by Art Spiegelman and *A Hard Day's Night* by the Beatles, both available on CD-ROM from: The Voyager Co., 578 Broadway, Suite 406, NY 10012.

97. Speech recognition and other advanced technologies may be added for appropriate situations.

Two main problems appear with the emerging Information Space:

- How will people live in an environment where they are reachable independent of time and place? Do people need constant information access? How will they cope with the mass of information available?
  One answer may be deferred communication, i.e. messages of any kind can be stored and read later. So, one can decide when to communicate and when to rest. There will also be support from intelligent agents that select relevant information on a personal basis.
- What will happen when the devices that people rely on will have access to very personal data? How can security and privacy be preserved?
  It will certainly be difficult, if not impossible, to guarantee security and privacy if personal data is stored centrally. Instead, personal data will have to be stored locally[98]. This is not a problem, because the structure of the global networks is massively distributed and there will be efficient ways to locate data anywhere.

When it comes to using contents produced by others there is also a serious copyright problem involved and there are no solutions yet for avoiding a mass of lawsuits in the future.

The discussion is open now, but the answers will only be ready when the technology is advanced and spread enough for the Information Space to become reality. It remains to be seen how people will deal with living in the Information Space – it will primarily depend on people's ability to make positive use of the new tools.

9 The Author's Final Word

This paper reflects the personal view of the author. Therefore, this work is intended to set the scope for further discussion rather than providing the reader with ultimate solutions. With change as the only constant parameter in the field of Pen Computing there is a need for ongoing discussion in which the author enjoys to take part. Comments, additions, and corrections are heartily welcomed and may be directed to: André Meyer, CS Life., Tessinerplatz 5, P.O. Box 497, CH–8021 Zürich, Switzerland, phone: +41 1 284 22 67, fax: +41 1 284 22 09, email: ameyer@ifi.unizh.ch.

Acknowledgments

First of all, my wife Chanie earns respect for her enduring support. Professor P. Stucki made this work possible thanks to his great trust and support. Thanks go to Martin Dürst of the MultiMedia Laboratory and Simcha Spielmann for proofreading and commenting, as well as to Jean-Renard Ward of Slate, Corp., for his valuable references, and various people from companies that provided me with information on their products: Heinz Sidler (NCR), El Saghe (IBM), Alain Ferru (Ferru Software), Pete Magowan (ARM), Kent Sandvik (Apple PIE), Hannes Speidel (Industrade), and Steve Jarrett (General Magic). An invaluable source for information was the Internet community from where I drew much of my insight. Last, but not least, I owe thanks to Mario Neuhäusler who enabled me to continue working on this paper while being occupied with lots of other work.

[98] On a PCMCIA card in a Personal Digital Assistant, for example.

References

Adorno, Theodor W., *Ästhetische Theorie*, Suhrkamp, Frankfurt am Main, 1970 (unfinished manuscript), ISBN 3–518–27602–6

Alford, Roger C., *The PCMCIA Redefines Portability*, BYTE, December 1992, pp. 237–244

Allen, Dennis, 1: *Pen-Input Systems*, BYTE, April 1992, p. 10; 2: *The Promise of High-Tech*, BYTE, September 1993, p. 10; 3: *Who Will Define the PDA?*, BYTE, October 1993, p. 10; 4: *Beyond the GUI*, BYTE, February 1994, p. 10

Amsoft, *Information for Software Development on the Amstrad PenPad PDA600*, available from: Amsoft PDA Peripherals, P.O. Box 111, Macclesfield, Cheshire, United Kingdom, phone: +44 625 572779, fax: +44 625 572894

Andrews, Dave, *Behind the Wheel of the First Zoomer and Newton PDAs*, BYTE, September 1993, pp. 22–26

Apiki, Steve, *A Unified Ink Standard*, BYTE, September 1993, p. 26

Apple PIE, 1: *A Guide to the Newton Toolkit*; 2: *The NewtonScript Programming Language*; 3: *Newton Programmer's Guide*; 4: *Newton BookMaker User's Guide*, Newton Toolkit (NTK) documentation, available with the NTK from: Apple Programmers and Developers Association (APDA), Apple Computer, Inc., P. O. Box 319, Buffalo, NY 14207–0319, phone: (800) 282 2732 (U.S.), (800) 637 0029 (Canada), (716) 871 6555, fax: (716) 871 6511, email: APDA@applelink.apple.com, ©Apple Computer, 1994

Apple, 1: *Dylan – an object-oriented dynamic language*, Apple Computer, Eastern Research and Technology, 1992, available from: Apple Computer, 1 Main Street, Cambridge, MA 02142, phone: (617) 374 5300, fax: (617) 374 5353 or send name and address information to dylan-manual-request@cambridge.apple.com, more information is available from http://cambridge.apple.com/pub/dylan; 2: *OpenDoc – Shaping Tomorrow's Software (White Paper)*, available from: Apple Computer Higher Education Gopher Server

ARM, *ARM – shaping the future* and *ARM610/710 RISC Processor Data Sheets*, all ©1993, available from: ARM, Advanced RISC Machines Limited, Fulbourn Road, Cherry Hinton, Cambridge, CB1 4JN, England, phone: +44 (0) 223 400400, fax: +44 (0) 223 400410, email: armltd@armltd.co.uk

Arnbak, Jens C., *The European (R)evolution of Wireless Digital Networks*, IEEE Communications, September 1993, pp. 74–82

AT&T, *Hobbit*, ©1992, available from: Personal Communication Systems, AT&T Microelectronics, 4995 Patrick Henry Drive, Santa Clara, CA 95054, phone: (800) 372 2447, fax: (215) 778 4106

Ballard, Nigel, *Nigel's PDA comparison chart*, current version available from: comp.sys.pen, CompuServe Newton and Palmtop forums, or from nigel@dataman.demon.co.uk

Baran, Nicholas, 1: *LCDs and Beyond*, BYTE, February 1991, pp. 229–236; 2: *Rough Gems: First Pen Systems Show Promise, Lack Refinement*, BYTE, April 1992, pp. 212–222; 3: *Zinc-Air Batteries: Long May You Run*, BYTE, May 1992, p. 28; 4: *The Outlook for Pen Computing*, BYTE, September 1992, pp. 159–164; 5: *Competition for Active Matrix*, BYTE, January 1994, p. 24; 6: *Object Operating Systems – Alliances Could Mean Better Integration*, BYTE, March 1994, p. 38

Baran, Nicholas, Dan Bricklin, Jeff Dao, Ken Dulaney, Howard Eglowstein, Steve Liffick, Kevin Mankin, Andy Reinhardt, *The Future of Pen Computing*, Roundtable, BYTE, March 1992, pp. 115–118

Baran, Nicholas, Dan Bricklin, Jeff Dao, Neil Katz, Steve Liffick, David Mentley, *The Future of Pen Computing, Part 2*, Roundtable, BYTE, April 1992, pp. 99–102

Barr, Christopher, Michael Neubarth, *Pen Pals*, PC Magazine, October 12, 1993, pp. 116–182

Barthes, Roland, 1: *Le degré zéro de l'écriture*, Seuil, Paris, 1953; 2: *Mythologies*, Seuil, Paris, 1957 (transl. *Mythen des Alltags*, Suhrkamp, Frankfurt am Main, 1964, ISBN 3–518–10092–0); 3: *L'empire des signes*, Albert Skira, Genève, 1970 (transl. *Das Reich der Zeichen*, Suhrkamp, Frankfurt am Main, 1981, ISBN 3–518–11077–2)

Bell South, *Simon Says...*, documentation from: BellSouth Cellular Corp., 1100 Peachtree Street, N. E., Suite 14D05, Atlanta, GA 30309, phone: (404) 249 0476 (Nicole Lipson), fax: (404) 249 0452

Betz, David, *Life with NewtonScript*, BYTE, March 1994, pp. 191–194

Bohländer, Egon, Walter Gora, *Mobilkommunikation – Technologien und Einsatzmöglichkeiten*, Datacom, Bergheim, 1992

Bollay, Dennison, *Dylan (dynamic language) or CLOS* +-, Journal of Object-Oriented Programming, November/December 1992, pp. 75–77

Bowen, Tamme D., Kelly M. Hall, *Towards a Better Understanding of Dylan*, Laboratory for Applied Logic, University of Idaho, Research Report LAL–93–01, May 1993

Brettschneider, Joachim, *Spracheingabe und Sprachausgabe: Technologien mit weitreichenden Einsatzmöglichkeiten*, Computerworld Schweiz, October 25, 1993, pp. 7–11

Bricklin, Dan, *Jot Defines Electronic Ink*, BYTE, October 1993, p. 110

Bruce, Richard, Scott Elrod, *The Electronic Whiteboard*, BYTE, July 1992, p. 166–167

Carr, Robert M., *The Point of the Pen*, BYTE, February 1991, pp. 211–221

Carr, Robert, Dan Shafer, *Power of PenPoint*, Addison-Wesley, Reading MA (etc.), 1991, ISBN 0–201–57763–1

Chang, Robin L. P., Theodosius Pavlidis, *Fuzzy Decision Tree Algorithms*, Fuzzy Models for Pattern Recognition – Methods That Search for Structure in Data, IEEE Press, New York, 1991, pp. 321–325

Chomsky, Noam, 1: *Syntactic Structures*, Mouton, Den Haag, 1957; 2: *Language and Mind*, Harcourt Brace, New York, 1968

CIL, Component Integration Laboratories, information on *OpenDoc* available via email: cil@cil.org

Clarkson, Mark, 1: *Objects and PenPoint*, BYTE, December 1992, pp. 158–159; 2: *All-Terrain Networking*, BYTE, August 1993, pp. 111–116

Crane, Hewitt D., Dimitri Rtischev, *Pen and Voice Unite*, BYTE, October 1993, p. 98–102

Crystaloid, *Liquid Crystal Displays: A Primer*, ©1993, available from: Crystaloid Electronics Co., 5282 Hudson Drive, Hudson, Ohio 44236–3769, phone: (216) 655 2429, fax: (216) 655 2176

Dan Kimura, Takayuki, *Hyperflow: A Visual Programming Language for Pen Computers*, Proceedings of the 1992 IEEE Workshop on Visual Languages, Seattle, Washington, IEEE Computer Society Press, Los Alamitos, CA, ISBN 0–8186–3090–6, pp. 125–132

Davidson, Clive, *The man* [Alan Kay] *who made computers personal*, New Scientist, 19 June 1993, pp. 32–35

Davis, Dwight B., *Turmoil In Pen Operating Systems*, Datamation, December 19, 1993, pp. 20–28

De Bruyne, Pieter, *Compact Large-Area Graphic Digitizer for Personal Computers*, IEEE Computer Graphics and Applications, December 1986, pp. 49–53

De Bruyne, Pieter, Robert Korolnik, 1: *Measurement of Static and Dynamic Calligraphic Features*; 2: *Segmentation and Recognition of Calligraphic Text*, Laboratory for Communication Technology, ETH Zentrum, CH–8092 Zürich, Switzerland

De Mori, Renato, Pietro Laface, *Use of Fuzzy Algorithms for Phonetic and Phonemic Labeling of Continuous Speech*, Fuzzy Models for Pattern Recognition – Methods That Search for Structure in Data, IEEE Press, New York, 1991, pp. 299–310

Department of Transportation, *GPS Navstar – User's Overview*, ©1991, available from: U.S. Department of Transportation, United States Coast Guard, Commanding Officer Global Positioning System Information Center (GPSIC), Omega Navigation System Center, 7323 Telegraph Road, Alexandria, VA 22310–3998, phone: (703) 313 5900, fax: (703) 313 5920

Dipert, Brian, Lou Hebert, *Flash memory goes mainstream*, IEEE Spectrum, October 1993, pp. 48–52

Dunne, Shane, *Towards Interactive Pen Input of Visual Languages*, Proceedings of the 1992 IEEE Workshop on Visual Languages, Seattle, Washington, IEEE Computer Society Press, Los Alamitos, CA, ISBN 0–8186–3090–6, pp. 243–245

Eberhardt, Reinhold, Walter Franz, *Mobilfunknetze – Technik, Systeme, Anwendungen*, Vieweg, Braunschweig/Wiesbaden, 1993, ISBN 3–528–06526–5

Eco, Umberto, *La struttura Assente*, Valentino Bompiani, Milano, 1968 (transl. *Einführung in die Semiotik*, Wilhelm Fink, München, 1972, ISBN 3–7705–0633–2)

Eldridge, James, *Filing in a flash*, IEEE Spectrum, October 1993, pp. 53–54

EO, *EO Personal Communicator* and other documentation, available from: EO Europe, Abberley House, Granhams Road, Great Shelford, Cambridge, CB2 5LQ, United Kingdom, phone: +44 (0) 223 843131, fax: +44 (0) 223 843032

Evins, Mikel, *Objects without Classes*, IEEE Computer, March 1994, pp. 104–109 (

Feuerstein, Martin J., Theodore S. Rappaport, *Wireless Personal Communications*, Kluwer Academic Publishers, Boston/London/Dordrecht, 1993, ISBN 0–7923–9280–9

Földes-Papp, Károly, *Vom Felsbild zum Alphabet: die Geschichte der Schrift von ihren frühesten Vorstufen bis zur modernen lateinischen Schreibschrift*, Belser, Stuttgart, 1987, ISBN 3–7630–1266–4

Foley, James D., Victor L. Wallace, Peggy Chan, *The Human Factors of Computer Graphics Interaction Techniques*, IEEE Computer Graphics and Applications, November 1984, pp. 13–48

Forman, George H., John Zahorjan, *The Challenges of Mobile Computing*, IEEE Computer, April 1994, pp. 38–47

General Magic, information package on *Magic Cap, Telescript*, and company background available from: General Magic, 2465 Latham Street, Mountain View, CA 94040, phone: (415) 965 0400, fax: (415) 965 9424, email: steve_jarrett@genmagic.com

Getting, Ivan I., *The Global Positioning System*, IEEE Spectrum, December 1993, pp. 36–47

Glinert, Ephraim P., *Visual Programming Environments – Paradigms and Systems / Applications and Issues*, IEEE Computer Society Press, Los Alamitos, 1990, ISBN 0-8186-8973-0 / 0-8186-8974-9

GO, 1: *Penpoint User Interface Design Reference*, Addison-Wesley, Reading MA (etc.), 1991, ISBN 0–201–60858–8; 2: *Penpoint Application Writing Guide*, Addison-Wesley, Reading MA (etc.), 1992, ISBN 0–201–60857–X; 3: *PenPoint Operating System: A Catalog of Products and Services* and other documentation, available from: GO Corporation, 919 East Hillsdale Boulevard, Suite 400, Foster City, CA 94404

Goldberg, Adele, David Robson, *Smalltalk-80 – The Language*, Addison Wesley, Reading MA (etc.), 1989, ISBN 0–201–13688–0

Goldberg, David, Cate Richardson, *Touch-Typing With a Stylus*, INTERCHI '93 Conference Proceedings, Amsterdam, 1993, pp. 80–87 or from ftp://parcftp.xerox.com/pub/unistrokes/InterCHI.ps

Goodisman, Aaron, *A Stylus-Based User Interface for Text: Entry and Editing*, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology and Xerox Corp., Palo Alto Research Center, 1991

Gore, Andrew, Mitch Ratcliffe, *Newton's Law*, Random House, New York, 1993, ISBN 0–679–74647–1

Gore, Andrew, *Telescript: The net is the platform*, MacWeek, 01.10.94, pp. 1, 95

Guyon, Isabelle, Lambert Schomaker, Réjean Plamondon, Mark Liberman, Stan Janet, *UNIPEN project of on-line data exchange and recognizer benchmarks*, Pattern Recognition and Neural Networks chapter of the International Conference on Pattern Recognition, Jerusalem, Israel, October 9–13, 1994, available from ftp://ftp.cis.upenn.edu/pub/UNIPEN-pub/

Halfhill, Tom R., 1: *PDAs Arrive – But Aren't Quite Here Yet*, BYTE, October 1993, pp. 66–86; 2: *Ink vs. ASCII*, BYTE, October 1993, p. 74

Halfhill, Tom R., Andy Reinhardt, *Just Like Magic?*, BYTE, February 1994, pp. 22–23

Hallett, Joe, *Monochrome to Color*, BYTE, July 1992, p. 179–182

Hayes, Frank, *Displays – Down the DRAM Drain?*, BYTE, February 1991, pp. 232–233

Holtzman, Jack M., David J. Goodman, *Wireless Communications – Future Directions*, Kluwer Academic Publishers, Boston/London/Dordrecht, 1993, ISBN 0–7923–9316–3

Horrocks, R. J., R. W. A. Scarr, *Future Trends in Telecommunications*, Wiley, New York, 1993, ISBN 0–471–93724–X

Huang, X. D., Y. Ariki, M. A. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, Edinburgh, U.K., 1990, ISBN 0–7486–0162–7

IBM, 1: *IBM ThinkPad 710T*; 2: *IBM ThinkPad 750P*; 3: *IBM TouchMobile Handcomputer System*, IBM Product Information

IEEE, *Visual Programming*, special edition of IEEE Computer, March 1995

Imielinski, Tomasz, B. R. Badrinath, 1: *Querying in highly mobile distributed environments*, Proceedings of the 18th VLDB Conference, Vancouver, Canada, 1992, 2: *Replication and Mobility*, 3: *Data Management for Mobile Computing*, 4: *Mobile Wireless Computing: Solutions and Challenges in Data Management*, DataMan Project, Rutgers University, all available from ftp://paul.rutgers.edu/pub/badri

Impedovo, S., J. C. Simon (editors), *From Pixels to Features III – Frontiers in Handwriting Recognition*, Elsevier Science, 1992, ISBN 0–444–89665–1

Jensen, Hans, *Die Schrift in Vergangenheit und Gegenwart*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1969, ISBN 3–326–00232–7

Khanide, Fereydun, 1: *Unleserlich – Wie gut erkennen Pen-Computer Handschriften?*, c't, April 1993, pp. 82–88; 2: *In eigener Schreibe*, c't, February 1994, pp. 110–120; 3: *Little Cards, Big Future*, BYTE European Reseller, March 1994, pp. 53–56

Kim, Joonki, *On-Line Gesture Recognition by Feature Analysis*, Computer Vision and Shape Recognition, World Scientific, Singapore (etc.), 1989, ISBN 9971–50–862–1, pp. 209–223

Knaster, Barbara, *Presenting Magic Cap*, Addison-Wesley, Reading MA (etc.), 1994, ISBN 0-201-40740-X

Kobb, Bennett Z., *Personal Wireless*, IEEE Spectrum, June 1993, pp. 20–25

Koblo, Martin, *Die Entwicklung der Schrift*, Brandstetter, Wiesbaden, 1963

Kolli, Raghu, Ralph Stuyver, Jim Hennessey, *A Conceptual Device for the Early Phase of Design*, Faculty of Industrial Design Engineering, Delft University of Technology, Jaffalaan 9, 2628 BX Delft, Netherlands, email: r.stuyver@io.tudelft.nl

Korolnik, Robert, *Erkennung handgeschriebener Zeichen*, Laboratory for Communication Technology, ETH Zentrum, CH–8092 Zürich, Switzerland, 1990

Krzyzak, Adam, W. Dai, Ching Y. Suen, *On the Recognition of Handwritten Characters Using Neural Networks*, Pattern Recognition – Architectures, Algorithms & Applcations, World Scientific, Singapore (etc.), 1991, ISBN 981–02–0604–6, pp. 115–135

LaLomia, Mary J., *User Acceptance of Handwritten Recognition Accuracy*, Conference Companion, CHI '94 (ACM Conference on Computer-Human Interaction), Boston, 1994, p. 107

Lechner, Bernard J., *HDTV is Coming to Desktop*, BYTE, July 1992, p. 189–192

Lévi-Strauss, Claude, 1: *La pensée sauvage*, Plon, Paris, 1962 (transl. *Das wilde Denken*, Suhrkamp, Frankfurt am Main, 1973, ISBN 3–518–27614–X); 2: *Le cru et le cuit*, Plon, Paris, 1964

Lichtman, Moshe, *Pen User Interface – 9 Considerations*, PEN, July/August 1992, pp. 28–29

Linderholm, Owen, Steve Apiki, Michael Nadeau, *The PC Gets More Personal*, BYTE, July 1992, pp. 128–138

Lurie, Michael, *Color and Resolution*, BYTE, July 1992, p. 171–176

McCrone, John, *Computers that listen*, New Scientist, December 4, 1993, pp. 30–35

McKeehan, Julie, Neil Rhodes, *Programming for the Newton*, Academic Press, 1994, ISBN 0–12–484800–1

Meisel, William S., *Talk to Your Computer*, BYTE, October 1993, p. 113–120

Mello, John P. Jr., *Future Communications*, BYTE, August 1993, pp. 95–106

Mello, John P. Jr., Peter Wayner, *Wireless Mobile Communications*, BYTE, February 1993, pp. 146–154

Mezick, Dan, *Pen Computing Catches On*, BYTE, October 1993, pp. 105–112

Microsoft, *Microsoft Windows for Pen Computing Programmer's Reference*, Microsoft Press, 1992, ISBN 1–55615–469–0

Miley, Michael, *Agent Technology – The fine line between smart design and intelligent software*, MacWeek, 04.19.93, pp. 41–44

Minsky, Marvin, *The Society of Mind*, Simon & Shuster, New York, 1985

Motorola, *PowerPC: The New Generation of Computing* and *PowerPC 601/602/603 Microprocessors, Technical Summaries*, all ©1993, available from: Motorola Literature Distribution Center, P.O. Box 20912, Phoenix, Arizona 85036, fax: (602) 994 6430

Nadeau, Michael, *Enabling Technologies*, BYTE, July 1992, pp. 132–133

Nagel, David, *Software agents prepare to do their duty*, MacWeek, 04.19.93, p. 44

Nanda, Sanjiv, David J. Goodman (editors), *Third Generation Wireless Information Networks*, Kluwer Academic Publishers, Boston/London/Dordrecht, 1992, ISBN 0–7923–9218–3

Neisser, Ulric, Paul Weene, *A Note on Human Recognition of Hand-Printed Characters*, Information and Control, Vol. 3, 1960, pp. 191–196

Nouboud, Fathallah, Réjean Plamondon, *A Handwriting Interface to a Schematic Pseudocode Generator*, Pattern Recognition – Architectures, Algorithms & Applications, World Scientific, Singapore (etc.), 1991, ISBN 981–02–0604–6, pp. 301–310

Novobilski, Andy, *PenPoint Programming*, Addison-Wesley, Reading MA (etc.), 1992, ISBN 0–201–60833–2

Pal, Sankar K., Dwijesh Dutta Majumder, *Fuzzy Sets and Decisionmaking Approaches in Vowel and Speaker Recognition*, Fuzzy Models for Pattern Recognition – Methods That Search for Structure in Data, IEEE Press, New York, 1991, pp. 236–240

PCMCIA, *PCMCIA Backgrounder* and *Update – The PCMCIA News*, ©1993, available from: Personal Computer Memory Card International Association, PCMCIA Headquarters, 1030G East Duane Avenue, Sunnyvale, CA 94086, phone: (408) 720 0107, fax: (408) 720 9416; *PCMCIA Specifications*, available from ftp://csd4.csd.uwm.edu/pub/Palmtop/pcmcia.specifications

Perratore, Ed, *Sharp's Non-Newtonian PDA*, BYTE, October 1993, p. 69

Piersol, Kurt, *A Close-Up of OpenDoc*, BYTE, March 1994, pp. 183–188

Plamondon, Réjean, Ching Y. Suen, Marvin L. Simner (editors), *Computer Recognition and Human Production of Handwriting*, Proceedings of the Third International Symposium on Handwriting and Computer Applications, Montréal, July 1987, World Scientific, Singapore (etc.), 1989, ISBN 9971–50–665–3

Plamondon, Réjean, Ching Y. Suen, *Thinning of Digitized Characters from Subjective Experiments: A Proposal for a Systematic Evaluation Protocol of Algorithms*, Computer Vision and Shape Recognition, World Scientific, Singapore (etc.), 1989, ISBN 9971–50–862–1, pp. 261–272

Plamondon, Réjean, G. Leedham, *Computer Processing of Handwriting*, World Scientific, Singapore (etc.), 1990, ISBN 981–02–0408–6 (hardcover), ISBN 981–02–0520–1 (paper)

Pountain, Dick, 1: *Amstrad's Modest PDA*, BYTE, August 1993, pp. 161–164; 2: *PDA CPUs: New Form Demands New Functions*, BYTE, October 1993, pp. 80–81

Prince, Betty, Roelof H. W. Salters, 1: *ICs going on a 3-V diet*, IEEE Spectrum, May 1992, pp. 22–25; 2: *Battery basics*, IEEE Spectrum, May 1992, p. 24

Prophet, Andy, *PCMCIA*, PEN, July/August 1992, pp. 10–12

Quain, John R., *No Tabula Rasa: Handwriting*, PC Magazine, October 12, 1993, p. 176

Rahnema, Moe, *Overview of the GSM System and Protocol Architecture*, IEEE Communications, April 1993, pp. 92–100

Redfern, Andy, *Apple ARMs Itself*, BYTE, July 1992, p. 134

Reimer, John, *Memories in My Pocket*, BYTE, February 1991, pp. 251–258

Reinhardt, Andy, 1: *Touch-and-Feel Interfaces*, BYTE, February 1991, pp. 223–226; 2: *Building the Data Highway*, BYTE, March 1994, pp. 46–74; 3: *Managing the New Document*, BYTE, August 1994, pp. 91–104

Rekhter, Yakov, *AnArchitecture for Transport Layer Transparent Support for Mobility*, T. J. Watson Research Center, IBM Corp.

Riecken, Doug (editor), *Intelligent Agents*, Communications of the ACM (special edition), July 1994, pp. 18–147

Rogowitz, Bernice E., *Displays: The Human Factor*, BYTE, July 1992, p. 195–200

Ryan, Bob, 1: *Communications Get Personal*, BYTE, February 1993, pp. 169–176, 2: *The Wireless Factor*, BYTE, October 1993, p. 84; 3: *Dynabook Revisited with Alan Kay*, BYTE, February 1991, pp. 203–208; 4: *Processors for PDAs: Intel, AT&T, and AMD Continue the Chase*, BYTE, December 1993, p. 28

Ryan, Bob, Tom Thompson, *RISC Grows Up*, BYTE, January 1994, pp. 91–98

Salberg-Steinhardt, Barbara, *Die Schrift: Geschichte – Gestaltung – Anwendung*, DuMont, Köln, 1983, ISBN 3–7701–1454–X

Salmasi, Allen B., *Personal Communication Networks of the Future: CDMA Digital Cellular & PCN Developments*, Wireless Personal Communications, Martin J. Feuerstein, Theodore S. Rappaport (editors), Kluwer Academic Publishers, Boston/London/Dordrecht, 1993, ISBN 0–7923–9280–9, pp. 61–76

Scheele, Peter, *Mobilfunk in Europa*, Decker, Heidelberg, 1992, ISBN 3–7685–0992–3

Schenkel, Markus, I. Guyon, D. Henderson, *On-Line Cursive Script Recognition using Time Delay Neural Networks and Hidden Markov Models*, Institute for Signal and Information Processing, ETH Zürich, CH–8092 Zürich, and AT&T Bell Laboratories, Holmdel, NJ 07733

Schwarz, Hans-Ulrich, *Schnittmuster – Handschrifterkennung mit der Winkelschnitt-Analyse*, c't, November 1993, pp. 248–253

Shepard, Jeffrey D., *Lower the Voltage, Raise the Power*, BYTE, December 1992, pp. 137–141

Simon, Jean-Claude, *Off-Line Cursive Word Recognition*, Proceedings of the IEEE, 1992, Vol. 80, Issue 7, pp. 1150–1161

Siy, Pepe, C. S. Chen, *Fuzzy Logic for Handwritten Numeral Character Recognition*, Fuzzy Models for Pattern Recognition – Methods That Search for Structure in Data, IEEE Press, New York, 1991, pp. 321–325

Software Publisher's Association, *JOT – A Specification for an Ink Storage and Interchange Format, Version 1.0*, ©Slate 1993, available from: Software Publisher's Association Mobile and Pen special-interest group, phone: (202) 452 1600 ext. 336; technical support from: Slate Technical Support, Slate Corp., 15035 N. 73rd Street, Scottsdale, Arizona 85260, phone: (602) 991 6844, fax: (602) 443 3606, email: 75300,3236 (CompuServe)

Srihari, Sargur (editor, chairman), *IWFHR III*,Proceedings of the Third International Workshop on Frontiers in Handwriting Recognition, Buffalo, New York, ©1993 Center of Excellence for Document Analysis and Recognition, CEDAR, SUNY at Buffalo, email: hamilton@cedar.buffalo.edu

Statt, Paul, *Intel/VLSI Join the PDA Fray*, BYTE, January 1994, pp. 101–105

Steel, Raymond, *Personal Communication Networks*, Tutorial at: International Zurich Seminar on Digital Communications, Zürich, March 1994

Sternglass, Daniel, *The future is in the PC cards*, IEEE Spectrum, June 1992, pp. 46–50

Suen, Ching Y., Christine Nadal, Raymond Legault, Tuan A. Mai, Louisa Lam, *Computer Recognition of Unconstrained Handwritten Numerals*, Proceedings of the IEEE, 1992, Vol. 80, Issue 7, pp. 1162–1180

Suen, Ching Y., Marc Berthod, Shunji Mori, *Automatic Recognition of Handprinted Characters – The State of the Art*, Proceedings of the IEEE, Vol. 68, No. 4, April 1980, pp. 469–487

Sutherland, Ivan E., *Sketchpad – A Man-Machine Graphical Communication System*, AFIPS Conference Proceedings, 1963, pp. 2–19, reprinted in [Glinert]

Tappert, Charles C., Ching Y. Suen, Toru Wahakara, *The State of the Art in On-Line Handwriting Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, Vol. 12, No. 8, pp. 787–808

Thé, Lee, *Pen Hardware: Get Practical*, Datamation, December 19, 1993, pp. 30–35

Thompson, Tom, Tom R. Halfhill, Michael Nadeau, *Ease of Use Is Relative*, BYTE, October 1993, pp. 89–94

Tracht, Allen E., *Adapting laptops to data acquisition*, IEEE Spectrum, October 1993, pp. 45–47

Udell, J., 1: *Windows Meets the Pen*, BYTE, June 1992, pp. 159–168; 2: *ComponentWare*, BYTE, May 1994, pp. 46–56

Ungar, David, Randall B. Smith, *SELF: The Power of Simplicity*, in: Lisp and Sybolic Computation, 4, 3, 1991, Kluwer Academic; along with many other articles also available from http:// self.stanford.edu/

van Someren, Alex, Carol Atack, *The ARM RISC Chip – A Programmer's Guide*, Addison-Wesley, Reading MA (etc.), 1993, ISBN 0–201–62410–9

Wakahara, Toru, Hiroshi Murase, Kazumi Odaka, *On-Line Handwriting Recognition*, Proceedings of the IEEE, 1992, Vol. 80, Issue 7, pp. 1181–1194

Wang, P. S. P. (editor), C. Y. Suen *et al.*, *Character & Handwriting Recognition – Expanding Frontiers*, World Scientific, Singapore (etc.), 1991, ISBN 981–02–0710–7

Ward, Jean Renard, Barry Blesser, *Interactive Recognition of Handprinted Characters for Computer Input*, IEEE Computer Graphics and Applications, September 1985, pp. 24–37

Ward, Jean Renard, Debra Schultz, *Digitizer Renaissance*, BYTE, January 1993, pp. 251–260

Ward, Jean Renard, Mark J. Phillips, *Digitizer Technology: Performance Characteristics and the Effects on the User Interface*, IEEE Computer Graphics and Applications, April 1987, pp. 31–44

Wayner, Peter, 1: *Stretching the Ether*, BYTE, February 1993, pp. 159–165; 2: *Optimal Character Recognition*, BYTE, December 1993, pp. 203–210; 3: *Objects on the March*, BYTE, January 1994, pp. 139–150; 4: *Agents Away*, BYTE, May 1994, pp. 113–118

Weiser, Mark, *Some Computer Science Issues in Ubiquitous Computing*, Communications of the ACM, July 1993, Vol. 36, No. 7, pp. 74–84

Weissman, David, Allen H. Levesque, Richard A. Dean, *Interoperable Wireless Data*, IEEE Communications, February 1993, pp. 68–77

Wellner, Pierre, *Interacting with Paper on the Digital Desk*, Communications of the ACM, July 1993, Vol. 36, No. 7, pp. 86–96

Werner, Kenneth I., *The flat panel's future*, IEEE Spectrum, November 1993, pp. 18–26

Wittgenstein, Ludwig, *Tractatus logico-pilosophicus*, Suhrkamp, Frankfurt am Main, 1963, ISBN 3–518–10012–2 (especially the paragraphs 4.016–4.0641)

Woodward Sr., Ollie C., Tom Long, *Display Technologies*, BYTE, July 1992, pp. 159–168

Xerox, *Xerox introduces 'Liveboard'*, Xerox product information

Xu, L., A. Krzyzak, C. Y. Suen, *Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition*, IEEE Transactions on Systems, Man and Cybernetics, 1992, Vol. 22, Issue 3, pp. 418–435

Younse, Jack M., *Mirrors on a chip*, IEEE Spectrum, November 1993, pp. 27–31

Periodicals

BYTE, monthly, McGraw-Hill Inc., One Phoenix Mill Lane, Peterborough, NH 03458, phone: (603) 924 9281, fax: (603) 924 2550, ftp://ftp.uu.net, ISSN 0360–5280, subscriptions: BYTE Subscriptions, P.O. Box 552, Hightstown, NJ 08520

Double-Tap and *Frameworks*, Software Frameworks Association, 10062 Miller Ave., Suite202–B, Cupertino, CA 95014–3466, phone: (408) 253 2765, fax: (408) 253 2767, email: info@frameworks.org, FrameWorks@applelink.apple.com
NB: *Double-Tap* was cancelled after only two issues and the remaining SFA activities were taken over by MacTech Magazine (contact: Neil Ticktin, phone: (310) 575 4343, email: publisher@xplain.com)

IEEE Personal Communications – The Magazine of Nomadic Communications and Computing, IEEE Communications Society, IEEE Customer Service Center, P.O. Box 1331, Piscataway, NJ 08855-1331, phone: (908) 981 0600, fax: (908) 981 9667, email: member.services@ieee.org

interactions, quarterly, Association for Computing Machinery (ACM), ACM Member Services, 1515 Broadway, 17th Floor, New York, NY 10036, phone: (212) 626 0500, fax: (212) 944 1318, email: ACMHELP@ACM.org

PEN, Personal Electronics News, bi-monthly, PenWorld Inc., 761 Deep Valley Drive, Rolling Hills Estates, CA 90274, phone: (310) 377 7858, fax: (310) 377 8218, email: 71333,124 (CompuServe)

Pen-Based Computing, bi-monthly, Stylus Publishing, P.O. Box 876, Sandpoint, Idaho 83864–0876, phone: (208) 265 5286, email: nbaran@well.sf.ca.us, ISSN 1054–4011

Pen Computing Magazine, Nigel Ballard, Bournemouth, UK, email: Nigel@dataman.demon.co.uk

PDA Developers, bi-monthly, Creative Digital Systems, 293 Corbett Avenue, San Francisco, CA 94114, phone: (415) 621 4252, fax: (415) 621 4922, email: CDS.SEM@applelink.apple.com, 72722,3225 (CompuServe)

Special Interest Groups

The following Special Interest Groups or User Groups are of interest:

- SPA Pen Computing SIG. The Software Publishers Association Pen Special Interest Group has more than 90 members and meets several times a year to confer on pen computing issues. Kristen Teegarden, 1730 M Street NW, Ste 700, Washington, DC 20036, phone: (202) 452 1600, x336
- North Atlanta Newton User Group (NANUG), Mark Underwood, 3381 Claire Circle, Marietta, GA 30066, email: mau@aix3.ema.com, EMA.COMPSV (AppleLink), MarkU (NewtonMail, eWorld), newsletter: _protoReality (distributed electronically as text, MS Word, and Newton Book document)
- Newton Assist Club, Clive Girling, 1/1 Harbour Yard, Chelsea Harbour, London, SW10 0XD, Great Britain, phone: +44 500 639 866, fax: +44 81 569 2992, newsletter: Stylus
- Worldwide PenPoint Developer's Organization (WPDO), David Schachter, 801 Middlefield Road, #8, Palo Alto, CA 94301-2916, phone (after 10 am): (415) 328 7425, fax: (415) 328 7154, email: ds@netcom.com, 70714,3017 (CompuServe)
- Newton Owners of Switzerland and Europe (NOSE), Manuel Friedrich, Zürichbergstrasse 24, CH–8032 Zürich, Switzerland, fax: +41 1 860 04 16, email: Manuel@ezinfo.vmsmail.ethz.ch, mailbox: +41 31 911 67 47 (FirstClass)
- and many more…

Videotapes

There is a series of five videotapes available featuring David Schachter (WPDO) and the most prominent Pen Computing personalities. Details are available from David Schachter at ds@netcom.com or Silicon Valley Teleconference Services, attn: Janet Morgan, phone: +1-408-255 6790, fax: +1-415-328 7154, email: JMorgan862@aol.com

On-line Information Sources

Much of the information that is interesting in the rapidly evolving field of Mobile and Pen Computing is available electronically. The Internet is the first address for electronic information, but commercial networks – e.g. eWorld, CompuServe, America Online – also offer a great deal of information.

Usenet Newsgroups, CompuServe Forums

- Relevant newsgroups on the *Internet* are – with new ones coming up frequently:
  Comp.sys.pen, Alt.periphs.pcmcia, Alt.toys.hi-tech, Comp.ai, Comp.ai.fuzzy, Comp.ai.neural-nets, Comp.fuzzy, Comp.groupware, Comp.human-factors, Comp.lang.dylan, Comp.lang.visual, Comp.neural-nets, Comp.object, Comp.os.geos, Comp.speech, Comp.std.wireless, Comp.sys.arm, Comp.sys.handhelds, Comp.sys.newton.announce, Comp.sys.newton.misc, Comp.sys.newton.programmer, Comp.sys.palmtops.
- Among the commercial *ClariNet Communications Corp.* newsgroups are of interest:
  Clari.nb.pen, Clari.nb.pda, Clari.nb.apple, Clari.nb.ibm, Clari.nb.telecom, Clari.nb.trends, Clari.tw.new_media, Clari.tw.telecom
- *CompuServe* forums of interest are:
  Pen Technology Forum (PENFORUM), Communication Intelligence Corp. (COMINT), GO Corp. (GOCORP), GRiD Systems (GRID), IBM ThinkPad Forum (THINKPAD), Microsoft Windows Extensions Forum (WINEXT), Newton/PIE Forum (NEWTON), PCMCIA (PCVENF), Pen Magazine (PENFORUM), Slate Corp. (SLATE), Software Publishers Association Forum (SPAFORUM), Telecommunication Issues Forum (TELECO)

Mailing Lists

Mailing lists can be subscribed electronically. News are then sent via email to all subscribers.

- *Edupage* news service: send email containing "sub edupage *firstname lastname*" to listproc@educom.edu
- *General Magic*: send email to listserv@brownvm.brown.edu containing "subscribe magiccap <name>" and/or "subscribe magicdev <name>"
- *GO PenPoint*: send email to penpoint@netcom.com
- *HOTT* list (hot off the tree) new service: send email containing "subscribe hott-list" to listserv@ucsd.edu
- *Innovation news service*: send email to Innovation-request@newsscan.com containing "subscribe"
- *NCR NotePad*: send email containing "subscribe notepad-list" to listserv@cerritos.edu
- *Newton Announce*: send email containing "subscribe newton-announce <email address>" to majordomo@oit.itd.umich.edu
- *Newton Berkeley Computer Society*: send email containing "subscribe bcs-newton" and/or "subscribe npc" (*Newton Programmer's Collective*) to majordomo@sunshine.mit.edu
- *Newton_News*: send email containing "subscribe newton_news" to listserv@netcom.com
- *Newton Medical List*: send email to listproc@dsg.harvard.edu containing "subscribe newtmed <name>"
- *OpenDoc* lists: send mail containing "lists" to majordomo@cil.org
- *Self*: self-interest@self.standford.edu, send email to self-request@self.standford.edu
- *Zoomer*: send email containing "subscribe zoomer-list *firstname lastname*" to listserv@grot.starconn.com

World-Wide Web (WWW): File (FTP) & Hypertext Transfer Protocol (HTTP)

- Newton Archive University of Iowa
  http://newton.uiowa.edu/
- Newton Directory of AMUG (NANUG)
  http://www.amug.org/amug_newton.html
- Stanford Newton User Group (SNUG)
  http://www.rahul.net/flasheridn/snug/
- Newton Archive Collection
  ftp://newton.sys.uea.ac.uk/Newton_Archive/
- Apple Newton Page
  http://www.apple.com/documents/newton.html
- Apple Newton Developer Info
  ftp://ftp.apple.com/dts/newton/
- Cadaver Newton
  ftp://cadaver.acm.ndsu.nodak.edu/pub/newton/
- Manitoba Newton
  ftp://ftp.cc.umanitoba.ca/Mac-Develop/Newton/
- NewtNews
  ftp://io.com/pub/usr/btorres/NewtNews/
- NewtReport
  http://www.iicm.tu-graz.ac.at/CNewtReport
- Newton Medical Applications
  http://med-amsa.bu.edu/newton.medical/newton.medical.html
- Lunatech
  http://www.lunatech.com/
- Innovative Computer Solutions
  http://www.indirect.com/ics/newton/
- PelicanWare
  http://www.teleport.com/~bettes/
- Apple Computer Home Page
  http://www.apple.com/
- Apple Cambridge (Dylan)
  http://www.cambridge.apple.com/
- Apple World Wide Web Home Page
  http://www.info.apple.com/
- Ziff-Davis (MacWeek, MacUser, etc.)
  http://www.ziff.com/
- PenWorld Magazine
  http://www.penworld.com/
- Pen-Based Computing Magazine
  ftp://ftp.netcom.com/pub/jerney/www/pbc.html
- infoLink Communication Services
  ftp://ftp.netcom.com/pub/il/ilink/infolink.html
- Educom
  http://educom.edu/
- Mark Weiser
  http://www.ubiq.com/hypertext/weiser/weiser.html
- Geos
  ftp://130.219.44.141/pub/geos/
- Zoomer
  http://www.eit.com/mailinglists/zoomer/zoomer.html
- General Magic
  http://wiretap.spies.com/ceej/magic.html
- Motorola Wireless Data Group (Marco & Envoy)
  http://www.mot.com/MIMS/WDG/
- Motorola PowerPC
  http://www.mot.com/PowerPC/

- Self
  http://self.stanford.edu/
- Xerox PARC
  ftp://parcftp.xerox.com/pub/mobile-ip/, ...unistrokes/
- Mobile and Wireless Computing Index
  http://snapple.cs.washington.edu:600/mobile/mobile_www.html
- UNIPEN
  ftp://ftp.cis.upenn.edu/pub/UNIPEN-pub/
- OpenDoc
  http://www.astro.nwu.edu/lentz/mac/programming/open-doc/home-od.html
- Taligent
  http://www.taligent.com/
- CSD Mobile Information
  ftp://csd4.csd.uwm.edu/pub/Portables/
- Delft University of Technology
  http://www.twi.tudelft.nl/Local/HCI/
- Rutgers University, Mobile Distributed
  ftp://paul.rutgers.edu/pub/badri/
- AT&T
  http://www.att.com/
- Palm Computing (Graffiti)
  ftp://ftp.netcom.com/pub/pa/palm/
- Mobile Office
  http://www.mobileoffice.com/
- HCI Launching Pad
  http://hydra.bgsu.edu/HCI/
- ACM SIGCHI
  http://www.acm.org/sigchi/
- ACM interactions
  http://info.acm.org/pubs/magazines/interactions/interactions.html
- Agents
  http://www.cs.umbc.edu/agents/
- Agent Info
  http://www.cs.bham.ac.uk/~amw/agents/index.html
- Practice & Future of Autonomous Agents
  http://www.ifi.unizh.ch/asi-aa.html
- MIT Agent Research (Media Lab)
  http://agents.www.media.mit.edu/groups/agents/