Ideas for future Pen products

Lloyd Frink: January 20, 1992

This purpose of this document is to generate ideas and discussion on what products the PenWin group might create in the future. It includes ideas and philosophy from many different people and places - it just so happens that I am writing them down here. As with any ideas, some are good, some are bad, and most are controversial. This document is a way to throw a bunch on the table.

A bit of Philosophy

Here is a quote from John Sculley at his Keynote speach at the recent Consumer Electronics show:

> Apple has always stood out as the contrarian in the computer industry, believing that ease-of-use, usefulness and user appeal were more important than faster performance, complex capabilities and added features. For us it's been a matter of deep beliefs that have made us passionate about the user's experience. What allowed us to succeed as the only company who hasn't adopted Microsoft's and Intel's technologies is that our users love their Macintoshes to the point of fanaticism. (1/9/92)

I think a lot of what he says could be applicable to PenWindows vs. GO, and evenutally Penwin vs. whatever Apple/Sony/General Magic do. When you talk to Mac users, most of them "love" their Mac. They really like showing it to their mother and grandmother. In some of the palmtop focus groups we have conducted, people are absolutely addicted to their Casio BOSS, Sharp Wizard, etc. I'd guess the palmtop users have this fervor mostly because of the new functionality provided by the device, not the elegance of the user interface provided. Soon, there surely will be clinics for those who are so addicted to Nintendo and Gameboy that they can't put them down.

People see Penwin, and they have this same kind of "wow" sense. But without a lot of "easy to use, useful, and appealing" pen-centric apps, this feeling might end up being rather fleeting.

So what to do? I think the emphasis in our future pen products should be on those products and features which create a more emotional attachment with the user. People like things which are intriguing, interesting, cool, and generally fun to use. They love games and being entertained. Artistic graphics, animation, and good metaphors make using apps a more pleasent experience. In addition to being easy to use and useful, which seem to be pretty standard design concepts we use at Microsoft, I think we need to put more emphasis on the third characteristic which Sculley points out - user appeal.

So how do we make our pen products more exciting? Well, the most important thing is coming up with lots of ideas from lots of people. Hence the purpose of this document - to throw out some ideas and hopefully generate more. I will be collecting and distributing a pile of other documents, papers, and news articles that will also help in this process. The goal of this collection is to provide people with a good background for ideas and for our "brainstorm month" after we ship.

What kind of compatibility is important?

Overall our primary accomplishment in V1 of penwin is to have created a solid foundation for creating new pen-centric applications. This goal got combined with *compatibility* for a number of reasons. In terms of usability, I think that trying to be compatible with existing unmodified Windows apps forced us to make some design sacrifices - primarily the way our gestures and overall UI work. In terms of development tools, being compatible with windows development tools, especially VB, has helped us a ton. For future pen products we should definititely rethink the costs and benefits associated with compatibility.

Here is one example of a tradeoff we made for compatibility - we made a zillon other smaller ones as well. All of our gestures are orthoganal to handwritten characters. We made this decision in part because existing applications never provide us with context to figure out when the user is doing data entry vs. gestures. It still may be the right thing to do for the future to have a modeless UI, but we are getting lots of feedback from users that say GO's gestures are more intuitive than our. In order to delete something in the GO model, you draw an 'X' over it. In Penwin you either do a delete, backspace, or cut which are all symbols people rarely if ever have drawn in their life. This is probably the most glaring example, but it shows the kind of trade-offs we have made in our design.

Even if we say compatibility means more than just running unmodified apps, the marketplace is associating compatibility with running unmodified apps. In addition to having made design sacrifices, running unmodified apps may turn out to be a weaknesses for two other reasons - (1) using unmodified apps will be more difficult and harder to use than people expect, many already complex apps being made MORE complex because a pen is being used, and (2) ISV's either doing nothing or very little to their existing windows apps and ISV's are also discouraged from doing new easier to use pen-centric apps.

And yet, our biggest win versus GO is the ability of our customers and ISV's to quickly get prototypes of their apps up and running. I don't think we know of one customer who is planning on using unmodified apps as part of their standard setup. The relative ease of development is due to the stability of the underlying OS, the wide availability of development tools, and the existing code base of Windows apps combined with the developers' knowledge of Windows programming. All of these compatability factors have played a large part in our success to date.

Point being, we should determine exactly which kinds of compatility - if any - will be important to us in our future products.

Target Markets for Pen Products

I think the market for Pen based devices can be broadly divided into two categories: consumer/mass market devices and buisness/higher end computers. Another deliniation of the two markets might be those products which cost around $500 and less, and those which are greater than $1000. Or low end, and high end machines. Right now we have a product, penwin 1.0, which targets the high end of the market. It is not clear that that this product or slight derivations of it (improved UI, better pen support, better HWX,

appropriate small shell and applets) meet the needs of this low end market. I.e. do you really want to run Windows on a little handheld machine?

I believe it is important to look at what people will actually do with both high end and low end devices. The most important thing to note about the list below is that many of them will be well suited for either a high end or low end machine. Here is a list:

Phone List/Rolodex, Calendar, To Do List, Notes
Sales Tool
    customer/contact List and relavent notes/action items, price list, product info, inventory status, e.g. personal finacial planners, Frito Lay salespeople, pharmecetical reps
Math paper (calculator/spreadsheet)
Messaging/Email : Person to Person Communication
Fax/Cellular
Games - Entertainment, FUN!
Voice Annotations
Voice Recognition
    small vocabulary, maybe just for commands and macros
Database Query Tool
Information Collection/Recording - Forms Filling Out
    insurance forms, meter reading, police tickets, sales force status, Fedex/UPS, traffic accidents, claim adjusters/damage assesments, cargo recording, building inspections, forest service data
Hospital Information for doctors, nurses, pharmicists - even paramedics
Mobil Order/Cash Register
    restaurant ordering, avis/hertz rental cars, walk up to people in Norstrom, lumber sales
Input device for Interactive TV
    Universal Remote Controller
Language Translator, Dictionary, Spell Checker, Thesarus
Information Publishing
    any magazine, newspaper, book. guide books, restaruants, movie reviews, entertainment guide.
Business Expense Tracking
Personal Finance Manager
    Checkbook, Budget, Expenses, Taxes, Income, Investments
Real Time Finanacial Information
    stock quotes, mutual funds, economic numbers, for brokers, traders, even clerks on trading floor
Wallet PC
    credit cards, electronic keys, identification, concert/sporting event tickets, child pictures
Project Scheduling
Maps, GPS locator, Traffic jam avoidance, parking space availability
Airline & Hotel Reservations, Places to Travel
Instant Photo Transmission
    with products like Logitech's Photoman, take pictures in field and transmit them on the spot: great addition for wireless communication

What Products Do We Do Next?

Kanji Pen Win

Goal: get to market quickly in Japan
Kanji Recogntion
mods to Kanji Win 3.0 and/or Kanji Win 3.1

Recognition

See Ericbe's document on Recongition plans.
Goal: Recognize a users natural handwriting
Incremental improvements to current technology - higher recognition rates, small resources
Next Generation - cursive, etc

Improve High End Product - PenWin 2.0

See Mikevk's document on Pen UI issues.
Goals: better support for pen-centric apps, more intuitive UI
improve UI and gestures
is complete existing app compatibility still a requirement?
improve ink support/editing
better compression
improved & new pen controls, support of existing controls for pen
better training/correction model
more development tools
need simple shell, task switcher
connectivity strategy
scalability solution

Create A Low End Product for Pen Palmtops

Goals: user appeal, simplicity, and usefulness

I think two fundamental assumptions can be made of users of pen palmtops: first, they will have had no experience with computers, and second, they will not read a manual. This implies the product has to be very simple to use.

Most of the horizontal users will buy a palmtop because they talked to a "fanatic" user. Isn't this what happened with Walkman's? Everyone saw other people walking around with them, they tried them out and decided to buy one for themselves. For a product to succeed in this manner it has to have strong consumer appeal and be very useful to that person.

To reiterate what I said above, what distinguishes palmtops from other pen computers is their cost, less than $500, and consequently their size, a palmtop. What I think we should do is create the best software we can to meets the needs of this low end market. This would include the underlying OS with a very simple UI, some useful startup apps, and some other very appealing applets and games.

Market Development

How do we go about getting part of this low end, palmtop market? Here are some thoughts on that.

I think our first key target customers are the obvious large consumer electronics manufacturers - Sony, Matsushita, Casio, Sharp, Seiko-Epson, Tandy, HP, Phillips, and maybe a few others who are making Pen machines now like NCR, Samsung, and Sanyo. Certainly there are other key names I have forgotten right now. When do we start approaching them? I think we should wait until we have more of a clue as to what general direction we are going to take - i.e. not right now. Should we approach just one or two "pilot" OEMs? Maybe several? Or take the Penwin approach and blanket the world? I kind of lean towards targeting several key OEMs, but leave out the zillion clone makers for the time being.

It would seem to me that we have a problem just selling them *Windows* and telling them it is an ideal OS for these palmtops. Not sure how plausible that is. I'll explain an analogy that might be similar to us going to all the consumer OEMs and telling them to build Windows palmtops. Imagine a car buyer looking for a nice, run of the mill four door sedan. He doesn't know a lot about brand names, but he has heard that Mercedes is a reliable car. So he walks into a Mercedes dealer and starts telling a salesman what he needs. Well, the only thing the salesman has is the low end, $20K, Mercedes 190E's. The customer says, no that's a bit to spendy for him, does Mercedes offer anything less expensive. And the salesman proceeds to do his job and try to convince the guy that he really *needs* all the nice features, solid reputation, high quality, etc that a Mercedes offers. Besides it is such a good investment, he'll actually *save* money because of the low depreciation, low repair costs, blah, blah, blah. Essentially, we might be in the same situation as this salesman in that we could very well be selling these OEMs and eventually their customers something they don't need or want. The real danger is that the OEMs will just write the software themselves or they will buy it from someone else like Apple/General Magic or maybe GO. I am most worried about Apple because they have a lot of resources, great connections with Sony, and a good reputation for creating exciting and innovative products. GO is probably a bit to defocused right now to be able to compete with Apple or the OEMs themselves. The big problem is we are not even in the game right now.

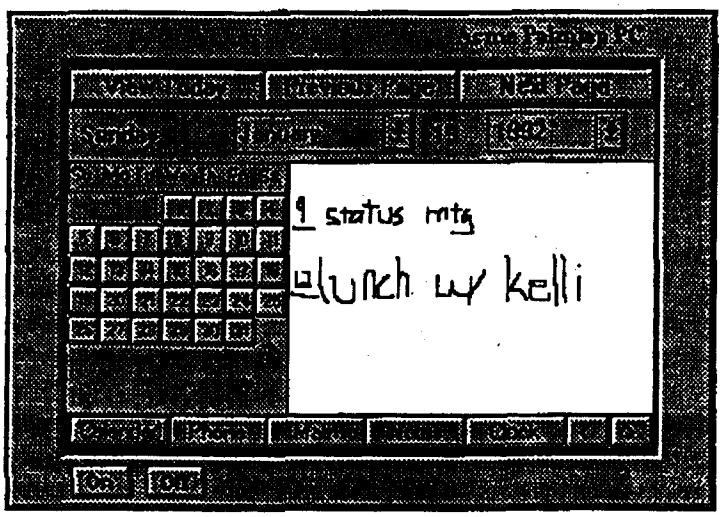
Two of the fundamental questions we have to answer soon are: (1) Which processor do we recommend? and (2) What is the underlying base for our palmtop OS? After we have come up with answers to these questions and have a vague notion of what the UI might look like, then I think we should start approaching OEMs. Until we make some tough decesions, it could be dangerous for us to get very detailed with them.

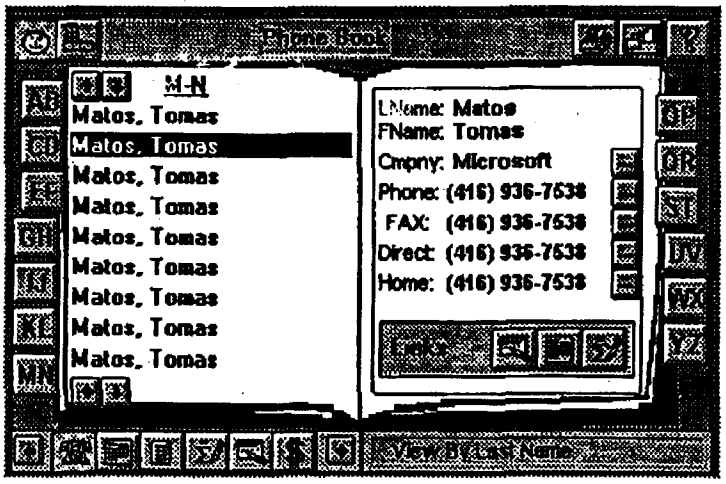
Different UI than Windows

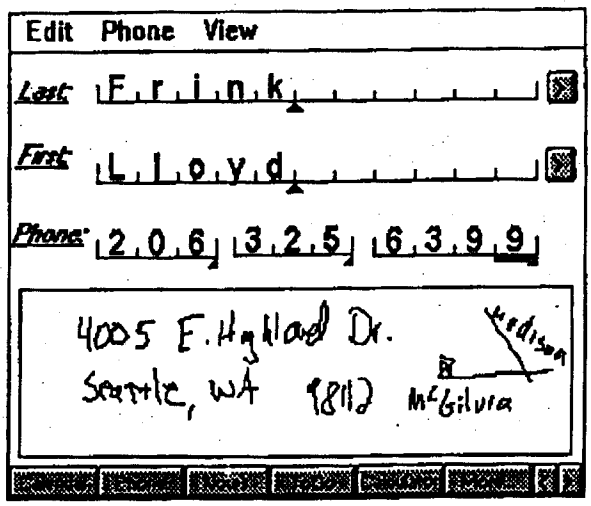
In order to make this palmtop OS very inutitive and accessible to non-computer users, we probably want to depart from some of the standard Windows user interface and DOS file system. We need to have a lot of discussion on what we should do, but here are a few examples. The key is that whatever we do might not look at all like Windows, it might not even run Windows apps because that is not what users want.

Some user interface and system behavior changes might include: no visible file system to normal users (i.e. no shell win 8.3 filenames and hierarchical directory structure), no overlapping windows, (!) only one maximize app visible at a time, no title bars, no minimize/maximize buttons, no scroll bars (maybe scroll buttons), concept of previous/next page inside of new apps.

Below are some idea screen shots. Mikevk did the first drawing, Tomasm did the second, and the third is mine. (All VB prototypes of varying functionality)







Target what processor?

Not Risc

We looked hard at possibly doing some sort of RISC version of Penwin and decided that it probably wouldn't be worthwhile for us to do. Maybe Nathanm would want to do a protable version of whatever he creates. The three factors that are most important are price, performance, and power consumption. After talking with Intel extensively and some of the RISC chip makers, we feel Intel is going to be close enough to the RISC guys on these three issues that it was not worth the cost of us trying to make Win 32 portable or do something from scratch.

However, we do seemed to have gotten a lot of attention from Intel by threatening to do a protable version of Windows. This has been useful in having them take us seriously in the palmtop market.

Probably 386 not 286, 386

Intel just said they are selling the 386SX at $40-$50 in quantity. They mentioned doing a 12 mHz SX for something like $20. While this may be a bit slow for Windows, it shows that the prices in the next year or two would make the 286 or 8086 a poor target for development.

Jump in bed with Intel

I think our strategy should be to continue working closely with Intel and get them to do a palmtop 386 chip (see document title Palmtop '386 chip by tomasm). This might be similar to us doing a subset of Win 32 without DOS. They have cancelled their HHC1 project which was 8086 PC-on-a-chip and are focusing on their HHC2 which is 386 based. We are actively trying to figure out exactly how serious Intel is in attacking this market.

Use what code base?

Win32 (without DOS 6)

Mackm and his group are open to getting rid of all real mode and DOS application support. We'd probably want to do a subset of all of the Win 32 API's and sacrifice application compatibility. We would want to add a lot of our own UI controls.

Overall, Win 32 looks like a good candidate, problem is we don't know when they will really ship and if we'll get bumped to the bottom of the priority list because they are selling 20 million copies of desktop Windows. We are having many fruitful discussions with Mack.

Win 16/DOS 5

This is not such a moving target because we know what it is and could likely get something out sooner. Questionable whether it would be worth all the effort updating this code base for things like scalability, connectivity, and in general optimizing for small resource machines.

DOS & VB DOS & CW

This is the slightly random idea that is worth at least considering. Instead of starting with this huge thing called Windows and ripping out the unneccesary parts, we might want to go the other way around. I.e. start with DOS and add what we need - use VB DOS as a development environemt for applets, add graphics support for VB DOS, use CW as a window manager, get some key components from GDI, and some sort of task manager. Would be really cool if all apps could be done in VB DOS and then they would work with relatively little work on VB Windows. Problem is that CW in graphics mode is said to have serious performance problems right now. Also would have to deal with segmented world of DOS and VB DOS.

Do Simple Shell/Task Switcher

We will have something besides the "Switch To..." and Task List which currently exists in Windows. This is probably all the shell most people would need - switch between apps. Like the "Button Bars" shown in the palmtop screenshots above. Any data storage is left up to the individul applications. They wouldn't use file names, or "save" and "load" of files. Whatever metaphor is most appropriate for themselves.

Do Startup apps, games, applets

Calendar/Schedule

This would have both a daily and a monthly view of ones schedule. It lets people keep track of and modify their appoinments. User information is left as ink. Ink is important because handwriting recognition of unconstrained handwriting is very difficult. The calendar/schedule is a very simple and straightforward implementation of what people would expect to see on a paper daytimer calendar.

The calendar/schedule is organized by date.

Phone List

This is a personal list of names and phone numbers. Only the first name, last name, and phone number of the person are recognized.

The phone list will be organized alphabetically according to a persons last name. One will be able to view both a list and card view of the phone numbers. The list view is simply a list box with each line containing a person's name and phone

number. In the list view one will be able to write any letters of the alphabet over the list and automatically scroll to that part of the list. Double tapping on a line will take the user to the card view. The card view will contain the recognized name and phone number as well as an ink field which allows the user to keep any information they want to about that particular person - address, phone extension, home phone, birthday, etc.

It should be remembered that this will be a simple yet very fuctional version of a phone list. Many other phone lists which have more recognized fields for other personal data will be written by ISVs and OEMs.

Notes

This will be a collection of a users notes. All of them will be in ink format. The key is that it be easy to enter the notes (just jot them down), and then easy to later find them. Hopefully we can do this with no handwriting recognition required. Ink editing will be very useful here.

The notes will be organized using a Table of Contents very similar to the PenWin notebook today. The creation and modification dates will also be automatically stored with each note.

Math Paper

I envision this to be something between a calculator and a spreadsheet. It will be an easy way for people to do calculations (mostly addition, subtraction, multiplication, and division) and keep track of them. The interface will be very direct and simple. Just write down what you want figured out and put and equals sign so that it gets calculated. You will be able to refer to other numbers on the current page and do calculations from them. If you change a number on the page, then all dependant results are recalculated. There will be no naming of cells or ranges, only direct references done by pointing. It will also be possible to use many different provided functions (mathematical and financial).

Since the user may want to keep his calculations around, and possibly annotate them, they could be part of the notes. This will be made possible by having the math paper be another layer, or mode, which can be used when inside of notes.

To Do List

Just a current list of things to do. Each item will have an associated priority, done/to be done check box, and ink description field.

To Do items will be organized by date and priority.

World Time

This will be a cool map of the world that allows you to point at any location and get the current time/date.

Categorizer

This will be way to *stamp* user defined keywords onto any of the above data types. It is provided as an additional way to organize ones data. Stamping is a way to attatch hidden words to a piece of information. It will be standard across all startup apps and is a resource which could be used by other apps as well. The idea is that a simple way to categorize information for later retrieval is to identify it by keywords.

The categorizer will be in the form of a simple dialog which is a series of drop down combo boxed edit lists boxes that have user contained keywords. For example a user could have a word in the first list which is "Company". When

"Company" is chosen, then the second list would then contain the names of all the companies the users has input. Possibly a third list could then contain divisions within each company. The idea is to make this a very free form way for a user to attatch keywords to information.

The information could later be found by having the user either write any keyword and search for it or by using the same categorizer dialog to create a find string.

The benefits of this method of categorization is that it lets the user avoid having to do handwriting recogntion and also have available a list of the keywords on which to search.

Voice Recognition

See Mikevk's document on Voice notes.

Goals: Start exploring market
Do our own voice recognizer?
Buy/liscense technology

Appendix A: More discussion of potential applications

Additional Palmtop Applications

Note: much of the following is from a previous document where I referred to the OS that we do for palmtops as *Tinywin*.

These are just a collection of some ideas for applications which could be developed for Tinywin palmtops. Because of its portability and compactness, the palmtop is not something which someone sits there and "uses" for long periods of time like a desktop computer or even a larger screen pen based notepads, but rather it is an accessory to what the person is trying to accomplish. The palmtop is not really the focus of attention, just a secondary tool which can be used in many situations.

Stock Quote, Finanacial Information

Wherever a person may be, they could receive the latest stock, bond, futures or other financial information. This could be used for real time trading purposes. Or, it could be used to update an individuals financial holdings.

The method for getting this information could vary depending on the needs of the customer. If it real time information was necessary at any location, the transport would be either radio or cellular. But if it were only within a specific location infrared might be used. And if a user simply wanted to update his information periodically, they might hook up to a phone outlet or some other sort of network tap.

Messaging/Email : Person to Person Communication

More and more communication in the future will be done via electronic mail. Many different types of data will be sent electronically. But how will palmtops fit into the world of desktop email, file/data transfer, and voice messaging?

Palmtop machines work well when *reading* text and ink email. Listening to voice mail would also be a good application. But viewing richer visual types of data created on larger color computer monitors probably would not be acceptable. As far a sending information, palmtops would be good for creating ink and voice mail, but not text mail (i.e. recognized handwriting) or message which contain more complex data.

I believe that palmtops will be a sufficient for many forms of informal person to person to person communication. And informal communication is most of what happens in the world. Most of these messages will be text and voice annotations, and in the future more handwriting and diagrams will be included because of the proliferation of pens as an input device.

Reading handwritten mail will, for at least several years, be easier for humans than for computers. (The reason computers may eventually catch up is because they can adapt to an individuals idiosycrasies in writing which may be totally confusing a person who has never seen the sender's handwriting before). It will be a challenge to display a handwritten note which has been written on a larger, higher resolution display than a palmtop. Zooming and scrolling the image are possible but not necessarily the best alternative. Algorithms which distinguish between text and diagrams will be used to accomplish this. It is not desirable to lower the resolution of handwriting because it will become illegible to the reader. But it probably is acceptable to lower the resolution of large diagrams so that the whole diagram can be seen on the screen. Being able to determine certain areas of ink to be words

objects will be helpful because word wrap can be used. All of this is dependant on the target resolution of the viewing device. If it is the same resolution, say VGA, but only on a smaller display, 3" x 5", then some of this shrinking may not be such a problem because none of the data is lost, it just is smaller with a higher DPI but still readable.

Voice Annotations

With adequate hardware, voice annotations will find their way into many applications. Primarily this will be through the use of Object Linking and Embedding (OLE) which means that each application will not have to support it explicitly.

The interface to this object would be similar to the existing OLE sound object. A user would choose to insert a sound annotation using either a gesture, button, or menu command. A dialog would appear on the screen which asked him to begin recording and he would push a button to start and stop. There would be a built in microphone on the palmtop to record the message. Listening to an object would simply mean double clicking on it. To edit the object a check mark gesture would be drawn on it.

One can see voice annotations put into email, notes, things to do and many other places. Palmtops equiped with voice annotation capabilities will have to have extra hardware to record and playback the sounds as well as sufficient memory capacity to store the annotations.

Database Query Tool

Many times when someone is in a meeting or out in the field they may want to perform a query on a remote database. More and more information is being put into databases, and it is important that people have access to it whenever they need it. The palmtop is an ideal way to accomplish this.

Here are several examples of how someone might use the query capabilities. If someone is sitting in a meeting and they want to find the sales of a product over the last several years, they could simply build a query, have it transmitted over a wireless network and then receive the result back almost instantly. Or if there is a salesman out in the field and they want to check the inventory and price of a certain item, they could easily do it. Maybe a lawyer would want to check a past testimony or a legal reference during a trial. Most of this queries will be just small bits of information which are helpful in the current situation. They are not likely queries which are the heart of a complex analysis. Those would be done in a more focused setting such as on a desktop machine.

Information Collection/Recording

People could use this to take inventory (with a bar-code reader) or when taking surveys. A person who reads energy meters could use a palmtop. Someone in a factory just checking on the status of different parts of the floor might find a palmtop useful. Or non-complex forms could be filled out on a small palmtop. Larger, more complex forms would likely require a pen based notepad.

Restaurant Ordering

When the waiter or waitress comes to your table, they have a small screen of buttons to push to make an order. Daily specials could pop up. If they ran out of a certain dish the waiter would be notified. After the order had been entered, it would be instantly transmitted to the kitchen. If it is a regular customer, maybe

their special requests and likely menu choices would be remembered. Sort of personalized service. Each restaurant could choose to make the program work with a finger or with a pen. The finger has the advantage of not having to worry about carrying around a pen. The pen has the advantage of being able to put more information on the screen because of the smaller buttons and also potentially having the waiter write down numbers (amount ordered) and scribbles for special instructions. A credit card reader could be built in as well to essentially be a mobil cash register.

Universal Remote Controller

This has many future possiblities. The first possibility is something that simply controls all existing infrared devices such as the TV, VCR, and stero. The buttons on the screen could be arranged, titled, and then "programmed" by the user. This application alone would not justify the price of a Tinywin palmtop because the same functionality can be bought for less than $100 today. But it would be a nice additional application of a palmtop which was bought primarily for some other purpose.

One can imagine when soon in the home the TV will be essentially a TV and a computer in one. All sorts of multimedia graphic and information retreivial applications will be available. Well, one will need some sort of an input device and a handy palmtop computer would be ideal. Using infrared technology, one could send and receive command choices from the TV. Most of the user interaction would just be pushing buttons with a finger, but say you did want to enter a number or have a word looked up you would just jot it down with the pen on the palmtop. Building queries to search remote databases and have the result shown on the TV would be nice. And just browsing around various information services also would be attractive.

Language Translator, Dictionary, Spell Checker, Thesarus

These are simply electronic versions of small books that people often carry around with them. Using handwriting recognition in boxes, which is relatively accurate, it would be much faster than looking up words. In addition to having a larger display than existing electronic word devices, a Tinywin palmtop would allow a more natural form of input, handwriting, than typing the word on a miniature keyboards. Again, most customers would not likely but a palmtop just to run one of these applications.

Information Publishing

These applications are a more general case of replacing books, manuals, and possibly even magazines. This is a concept similar to the Sony Data Discman. The important hardware requirement would be a large amount of read only storage.

It would be great to carry something around like the Microsoft Bookshelf. Or an insurance person might want to have replacement cost information availble for any part of the most common cars. This way they could give and insurance estimate on the spot. Even types of information published on a more periodic basis would be useful in a mobil environment such as an entertainment guide.

Business Expense Tracking

If a palmtop meant I didn't have to do expense reports, I could easily justify the cost of one. Needless to say, many people spend a lot of time doing expense reports. Service professionals who bill their time such as lawyers, accountants, and

consultants could use a palmtop to track their time and other expenses. This would be a popular and therefore ideal application for a palmtop. Each company's expense tracking is different, so a flexible application is necessary.

Personal Finance Manager

On the other hand, nearly everyone would like an easy way to easily keep track of their personal expenses and income. The problem with desktop solutions for personal finance is that a lot of spending is done away from a desk. The palmtop would make an ideal companion to these desktop applications. The palmtop could also serve as a standalone version to keep personal finance information.

Digital Money

This is the idea presented in Nathan's memo on consumer computers. Briefly stated one would be able to use a palmtop to replace their cash, checkbook, credit, and debit cards. This will be a popular application and will fit in well with the personal and business finance tracking described above. Some of the issues involved in making digital money possible are security and getting standards set amoung many different parties for electronic transfer of money.

General List Management

People often keep lists of many things. The most common such as phone numbers, things to do, and expenses deserve a specialized application of their own. But other lists such as sales calls, phone calls, or orders could be created with a general list mangement program. Each list would be a single "file" (i.e. not relational), and could have different user defined type of fields. This would be an application with the same functionality of a flat file manager, but targetted for small handwriting machines.

Games

A Tinywin palmtop would be a very high end version of the many game machines which are around today. Fingers would be used mostly to control the games. Both video action games and more "thinking" games such as Crossword puzzles would be popular.

Maps

A delivery person, business traveler or tourist could use a map program in a palmtop to navigate. Other information besides just street names and addresses could be combined with a map - e.g. business types, restaurants, movie theaters etc.

Project Scheduling

Project scheduling on a palmtop would be a companion product to a project scheduling program on a PC. One would be able to carry around information on a project, for example construction, and enter in any necessary changes. This information could then be updated either in real time or later on the main PC which has the full fledged scheduling program.

Sports Statistics

This is an application for the true sports enthusiast. Baseball fans could record every pitch and even look at past averages. Football fans might want to diagram plays. Coaches could find these devices useful for providing information during a game.

Embedded TinyWin Applications

These are application which could be built into existing larger devices that require a flexible interface to them. The first devices that will have TinyWin built into it are the most expensive ones where adding the hardware required for TinyWin is not a large percentage of the cost of the device.

Phone/Fax

A tinywin phone/fax would be the ultimate smart phone. The screen could appear exactly like a regular phone with the standard twelve buttons and traditional buttons such as *Hold*, *Transfer*, *Speed Dial*, etc. A finger would be used to operate these buttons.

Then many additional functions could be added. A person could write a scribble and send it to the person on the other end of the line. One could recieve a fax, then mark it up and resend it. All of a person's phone and fax numbers could be stored

Cars

Climate control, maps, trip setting, radio, CD player, cellular phone.

Planes

A zillion things in the cockpit.

Laboratory/Medical/Factory Equipment

Photocopy Machines

I have seen some fairly sophisticated panels used to control high end copy machines. These could be replaced by a Tinywin display.