InterOffice Memo

To: Jim Allchin, Steve Ballmer, Susan Boeschen, Mike Brown, Tom Corddry, Marlin Eller, Charles Fitzgerald, Lloyd Frink, Bill Gates, Frank Gaudette, Aaron Getz, Rob Glaser, Karen Hargrove, Edward Jung, Jonathan Lazurus, Dan Ling, Mike Maples, Paul Maritz, Peter Mollman, Cameron Myhrvold, Sanjay Parthasarathy, Rick Rashid, Greg Riker, Raleigh Roark, Darryl Rubin, John Sabol, Steve Shaiman, Brad Silverberg

From: Nathan P. Myhrvold

Date: September 8, 1992

Subject: **New Business Models for Wide Area Consumer Computing**

There is a class of consumer devices which have a *large retail business associated with charging people to use the device*, typically because it is connected to a *wide area network* which has per use or time based fees. Current examples include the telephone, FAX and cable TV, but our general expectation is that this area will expand enormously in the next five years. New devices of this sort will include smart telephones, intelligent FAX machines, pay-per-view and interactive TV and last but not least the expansion of on line information services such as Compuserve, America On Line or Minitel.

This class of device offers some interesting new opportunities for us to expand our current business model, or at least rethink our options. This memo discusses some ideas in this area.

Microsoft's Current Business Models

A brief review of Microsoft's current business models will help set the stage for the new areas. There are only four general models for Microsoft's really successful businesses in the PC industry, and each has some very specific characteristics. Here is a simplified list of the key features for each of them.

System Software

- End user value is delivered by ISVs. Any successful operating system does not directly deliver more than a small percent of the "value" the end user perceives, because if it is a success there will be a tremendous variety of application software. This is not to say that the system can't do something useful (it must) and this will include features which are directly accessible to the end user (bundled applets, shell, utilities...) but the real value is to be a platform for ISVs. Courting these ISVs is a critical aspect of system software strategy.
- Natural forces drive high market share. The value of system software to a user in most cases is proportional to the number of applications. The value of writing the applications to the ISV is proportional to the number of users. This feedback cycle drives rapid convergence toward a standard platform which historically tops out at 90% or higher market share.
- The basic revenue model is a small per system "tax". Since the share for a successful system is very high, the natural model is to consider the revenue as a percentage of the hardware system price. OEM agreements can make this very direct, but even if the system is distributed retail it amounts to the same thing. The

typical order of magnitude revenue is 1% of the system - this might go from 0.5% to 3%.

- **Complex interaction with ISV code is a barrier to entry.** The relationship of an application to the system APIs is similar to the relationship that the roots of a tree have with the ground - it is a very complicated and this makes it difficult for third parties to clone. This helps prevent competitors from dislodging a successful operating system.
- **Evolution and innovation provide another barrier as well as upgrade revenue.** The system must evolve its APIs and implementation over time in order to remain successful. This gives ISVs more features to exploit, makes it more difficult to clone, and it gives users a reason to pay for an upgrade.

There are many other interesting things that can be said about system software, but I think that this basically covers the aspects that are important to the business model.

Horizontal Application Software

- **Value is directly delivered in the code.** Applications must *do* something for people in a pretty direct fashion.
- **Basic revenue model is retail sales supplemented by upgrades.** The area in which Microsoft has concentrated is "horizontal" application categories which appeal to a large cross section of computer users, however even the highest categories only have about 60% share (as a category). The market share within a category is also smaller than the systems market - usually it tops out at 60% to 75%, so even the most successful applications only have a 40% - 50% share of machines. One can impute an effective per system revenue for an application, and this is useful as a comparative metric, but only in a statistical sense. The buyer of a new machine *must* have some sort of operating system, but it is certainly possible to avoid having a spreadsheet or database.
- **End user loyalty is maintained through UI and file format.** Once people learn an application and its data files they have a certain degree of loyalty to them. File import features and UI "help" facilities which translate commands are effective in breaching these barriers, so the net degree to which a user is "hooked" on an application is typically much less than the degree to which an application is "hooked" on a particular set of system software APIs.
- **Brand loyalty is important.** People are very aware of their application provider and this is an important ingredient to any successful product.
- **Regular upgrades are important for both revenue and loyalty.** A feeling of progress and improvement is necessary to keep users loyal to a product because it helps balance any desire to avoid the hassle of a competitive upgrade. This is also an important way to produce revenue (see below).

Accessories

My definition of an "accessory" is a product which is purchased primarily for use with another one of our software products, although it may have other uses. This includes both hardware (the Mouse, Softcard etc.) and software or data (TrueType FontPack). Accessories have some features of both applications and systems products as well as some of their own:

- **Low absolute cost, and software economics.** Our most successful accessories seem to be in the $50 to $150 range for end user prices, which makes them an impulse buy. Beyond that point, and it is not clear that you can keep the "accessory" status.

This is very relevant because accessories (as defined here) are basically a retail business. We have had some success at selling accessories like the Mouse on a bundled basis to OEMs, but I think that this is only because of the brand presence we established by having a strong retail business. Otherwise there is a strong price pressure to make the accessory a free bundle. Because they need a strong retail presence, accessories must be able to be sold through our distribution channels, and have a software-like cost of goods, even if they include hardware.

- Must be a companion to another strong Microsoft product. The key property of an "accessory" is that it works very well with another MS product, and that the two can be thought of as a complementary set. In the case of the Mouse, this was Word and later on, Windows, the FontPack was sold with Win 3.1 etc.
- Combined offers and bundles help get things established. Selling the accessory in the same SKU as the product it complements is one powerful way to promote it, but of course if you don't unbundle it later on it is not really an accessory. Including the accessory as a separately priced item during a direct sales offer, upgrade etc. will capture a lot of sales.
- ISV support helps a lot. Although it is important to sell an accessory in combination with one of our products, we also want to reach out to other ISVs for support. Historically speaking, we have been able to get by with less evangelism than we've had for system software, and with a much simpler interface. In the case of both the Mouse and the FontPack, the interface through which other ISVs take advantage of the accessory (at one level Windows, at another level the mouse driver or TrueType font format) is quite simple and is *not* a unique proprietary advantage in the same way that our system APIs are. The point here is that the accessory business can tolerate a much weaker level of ISV support and ISV lock in than system software.
- It is important to establish a brand identity and get in early. We have continued to be successful in the mouse business (and before that in the Softcard) in large part because we were there first.

As an aside, which is not relevant to the main purpose of this memo , I think that *every strong Microsoft product should have a $50 to $100 accessory product.* As a case in point, I think that Excel and Word could each support at least one extremely lucrative accessory.

Upgrades

The final business model for us is upgrades. This could be considered as part of the others, but upgrades have a number of general properties regardless of whether the original product is an application or systems software.

- Low price. Upgrades are typically limited to $50 to $100, or slightly more for competitive upgrades. Among other things this does not make it a very hard decision for the upgradee.
- Competitive upgrades allow you to milk your competitors installed base. Of course this cuts two ways since they can play the same game.
- Require a lot of end user visibility. People generally want to have the latest and greatest, but in order for this to work they have to be aware of what they have and what they should have. Direct mail and other intensive user awareness campaigns are needed.
- Regular frequency. Upgrades represent the closest thing we have to an annual fee or subscription. This is a powerful way to draw revenue from the installed base,

and to keep them loyal to our product. The more regular the frequency, the more we can get people in the upgrade habit, and have a sense they can depend on continual improvement.

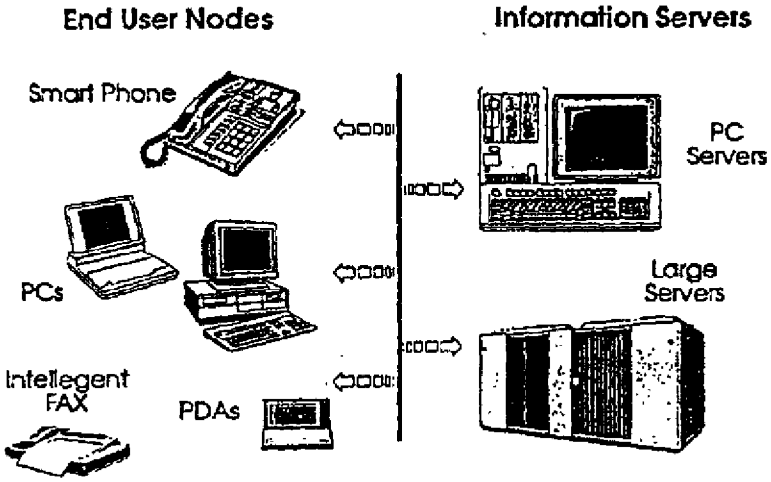
Wide Area Consumer Computing

Wide area consumer devices are things connected to a two way communication network. This includes the telephone, cellular phone, FAX machine or cable TV system (most are quasi-two way). In each case you pay a network provider who either pumps information at you, or allows you to pump it to somebody else. There is a fee associated with this transfer, which often dominates the cost of the "terminal node" or box - the physical device which is used to access the network. The general rule of thumb is an end user pays on-line costs which total between 100% to 1000% of the cost of the box *every year*. Since the typical box lasts 3 to 5 years, the bulk of the cost over the lifetime is the on-line component.

Not all consumer devices fit this model. Another set of devices is not connected to an on-line network, but also has the property that it consumes information. In these cases you buy recorded information for the box which you may reuse, but the typical turnover is such that users buy more on a regular basis - once again the information or use fee dominates the box price. Music players (CD, most tape players...) and video (Videodisk and most VCRs) each fall into this category. Nintendo and other home game systems do to, because there are trends and fads among game players. You could in principle play the same game or listen to the same CT for years, but most people wind up buying more albums, or renting tapes for their VCR or other things which amount to an ongoing revenue stream for usage. Although there are some similarities to the wide area case, we won't consider this class in this memo.

Television and radio are part of a wide area network, but it is only one way, and at least in the United States have no ongoing fee for normal use. Countries like Great Britain and Japan that have mandatory TV licensing could be considered as exceptions to this, but that does not help illuminate any points here. Cameras and camcorders do not fit because, like most uses of the PC, they are primarily *authoring* devices. People create their own information with them. There are some media charges associated with using them which may add up over time, but this is fundamentally different from consuming information or connect time.

The general picture for wide area consumer computing is as follows:



The node end is an intelligent device which is able to run the Remote Information Protocol or RIP, which was described in an earlier memo. RIP provides a way to send both information and GUI controls over a reasonably low bandwidth line. The servers provide information in the most general sense - it could be sports, news or weather, or narrow domain information such as Lexis or Nexis, bulletin board services or electronic software distribution. It could also provide an interface to control your finances and accounts, or to order physical items for later delivery from on line catalogs.

The examples in the picture above are primarily connected via the telephone line. Interactive cable TV is also part of the general vision. In that case the node is a smart cable box and connection between the node and the server is done through two way cable, or it uses the phone for the return path, or some other scheme such as those proposed by Interactive Networks or TV Answer.

A very key part of this vision is that we would like to have *organizations of any scale participate on either end.* A server application might be a large company such as an RBOC, Dow Jones or News Corp selling mass market information, Citibank providing you with banking services or someone like Land's End or L.L. Bean letting you do mail order shopping via a GUI interface. Some of these, such as the existing DowQuest service from Dow Jones, will use huge parallel computers to provide the necessary computational power to service complex requests, whereas others might need large transaction processing systems.

The analogy between personal computing and mainframe computing is very relevant here. Large servers will be important, but they can only deal with very large scale services which need to be centralized. Numerically speaking these large outfits *will only be a small fraction of the servers* - most of the servers in operation on any given day will be more like the corner deli or pizza parlor getting delivery orders, a doctor's office or hair salon scheduling reservations etc. The ultimate goal is that nearly any business which advertises in the Yellow Pages, or has a FAX machine should have some sort of server for remote access.

People sometimes suggest that the right approach is to have big companies act as the operators for servers - so that the small fry simply rent N megabytes on a giant server someplace. This might happen to some degree, but it can never be as flexible as having lots of PC based servers. The corner deli needs to see the sandwich orders at their location - not at a central server - and they need to have the receipts put into a form that PC applications can use for their accounting and other tasks.

This *scalable server strategy* is extremely important. It is what allows us to make the protocol pervasive, and it also gives us a huge advantage because *small servers are PCs.* People like General Magic, AT&T, News Corp, Time Warner and others who want to play in this area are, generally speaking, a lot less able to roll out a comprehensive PC based server software strategy than we are. In addition, the data which is sent out on the server, or that which is brought in must connect to a PC oriented world. The PC is the only way to democratize information services and make them truly ubiquitous and pervasive.

Another very important part of the vision is that *servers must be able to make money.* The only way for there to be lots of these things is to make it pay to have them. We must make the barrier for entry be extremely low so that nearly anybody can get in the server business, and do so profitably.

If we look at how to make money from this market, there are a number of very straightforward things to do.

PRODUCT	BUSINESS MODEL
Windows RIP front end	Bundled free with future Windows versions
Windows based server for PCs	Sold as an application - $195 - $495
NT based server for MP machines	Sold as an application - $495 - $995
RIP license for other servers	?
System software for consumer nodes	Typically .5% to 1% of node system price

In order to bootstrap the market, it is likely that we would bundle the RIP front end with Windows - hopefully both Cairo and Chicago. Cairo already has goals of having bundled connections to on line information services as part of the IAYF mission, and RIP would enhance the *presentation* of any such information.

Giving the PC nodes away removes one revenue source, but we can sell the scalable server software for both uniprocessor machines and large MP servers. This is a classic horizontal application, and the business model would be similar - you buy the server and away you go. Note that the "server" code in this case is actually a set of tools which let you easily customize or snap together an information server - a bulletin board construction kit.

This covers a very wide range of machines, but not everything. We would probably want to license the RIP protocol for use with machines other than NT servers. The amount we can charge depends on the strength of our intellectual property position. I do not see this as being a major revenue opportunity.

Finally there is the system software for the nodes - the operating system, shell and bundled apps for the smart phone, intelligent FAX machine etc. This is a classic systems business, and the progress made by the Digital Communications group would indicate that we can the same sort of royalties that we get in the PC business. This will range from $2 or so for smart telephones up to $25 or more for high end FAX machines.

Note that in order for the systems software model to really hold true, we have to have ISVs and the RIP protocol must include the ability to do download code. Unless we have a very complex hook of this sort, it is difficult to see how we could avoid being cloned. Also, without having ISVs to provide lots of unique value and explore lots of niches, we would lose one of the key success factors that helped drive the PC industry.

These straightforward approaches to the problem are all quite promising, but I do not believe they are the end of the story. The sections below describe other ways for us to make money on these machines.

PIMs & Node Applications

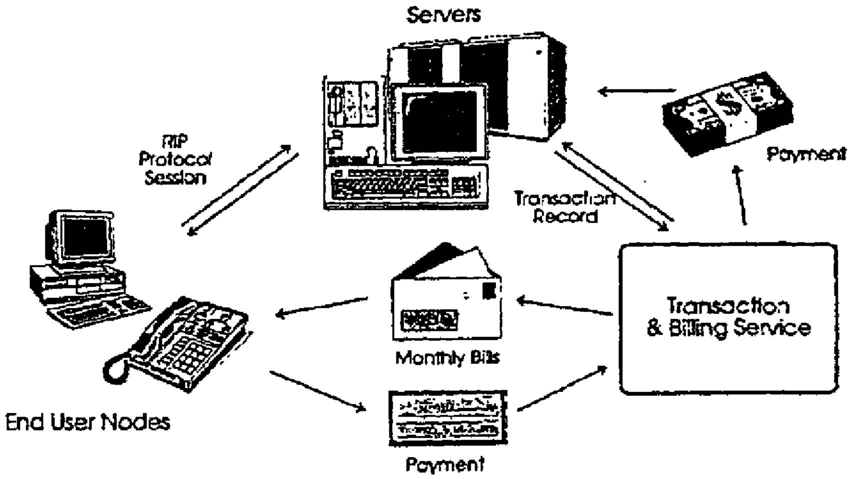
One possibility would be to create applications in the PC sense - i.e. something that you buy and install on the machine. This could include Personal Information Managers, games, small business applications, a special version of Microsoft Money, and a variety of other things which might make use of wide area connectivity, or might run stand alone.

Given the low cost of the nodes, the total amount you can charge for an application is probably going to be limited to $50, perhaps $100 for a *very* special one aimed at business. The bulk of the node application business will be in the $10 to $20 range. Electronic software distribution can make this very cost effective (you sell the software through a server).

Microsoft's role should focus on the high volume horizontal opportunities which can get reasonably large market share. We should clearly have node applications in mind, and investigate what the appropriate categories are. I regard this as the biggest challenge - I do not have any particularly unique ideas about node apps at this point.

Transaction & Billing Service

Essentially any wide area system which is used with for-profit servers must have some way for server operators to collect money. If you only have a few large servers this can be done directly by the server operator, which is the approach taken by Compuserve and Prodigy. This is impractical if you believe in scalable servers, because it is difficult for people with small server businesses to handle billing and collection directly. This suggests a system such as the one depicted below.



The RIP protocol session might deliver information directly, or it might order physical goods - either way you need to have a payment method, which is handled by the transaction and billing service. There are considerable economies of scale involved in operating the transaction and billing service. Historically this has been exploited by companies like American Express, Visa and others to provide a very similar service to companies that accept credit cards.

For-profit phone services have a similar set up. The billing is either handled by a long distance phone company (via 900 area code) or by an RBOC for the local calls (typically with a 976 or 970 prefix). In each case the billing is done as part of the normal phone bill. The transaction cost is much higher in this case because of the phone charges, and various other factors, including lack of competition from other payment means (credit cards must compete with cash or checks). The French Minitel system works on a similar basis.

Some general system of this sort *must* be put into place in order to create a widespread information marketplace. Each server "kit" should include the software and the necessary forms or account number which allows the purchaser to set up a connection to the billing service and go into business immediately - equivalent to the ability to accept a credit card or start a 900-number phone service.

I believe that this is actually a huge business opportunity for Microsoft, and I propose that we investigate forming such a transaction and billing service as a joint venture with a financial services company which have expertise in the mechanics of running this sort of operation. This may sound like an odd thing to suggest, but there are a number of things in its favor.

First, the existing services are not well suited to the task. Systems based on the network itself, such as 900 and 976 numbers, or the Minitel system are invariably time based, because that is the fundamental basis for the network. This is appropriate for a certain class of services (phone sex is popular on all three systems), but it is not the right model for ordering a pizza, or a gadget from the Sharper Image catalog, or a new software update from Microsoft. In those cases the call time is irrelevant because the transaction price depends on what you have ordered. With enough effort, the phone network people could get into the act, but only by creating a new service.

Credit cards are also not well suited at present. Current cards lack any form of electronic security, because the system was designed around using an original handwritten signature on a slip of paper. Today the transfer is all electronic, but the signature is still part of the verification procedure - both legally and practically. Mail order houses typically accept credit card orders, but it is not clear if the current system would survive the kind of pressure that widespread use for electronic billing would pose. It is not hard to fix this by using public key cryptography, either in a smart card or built into the node, but that would amount to creating a new service.

Another problem with credit cards is that they force people to use a transaction model in the strict sense of the word. When your credit card is run through the little machine by a cashier, it checks a central server to see if it is a bad card, and then it must lock the record and do a complete database transaction. A new digital system could use this method, but also has the option of using "digital money" - a digital equivalent of a traveller's check which does not require any on line check at transaction time. In this case the transaction fee can be much less. The server operator would exchange these digital certificates with the service for real money whenever they wanted to (daily or weekly) rather than on each transaction. The digital money would be "sold" by the service just as with paper based traveler's checks - which raises the opportunity to profit from the "float". American Express usually offers traveller's checks at a 1% fee and credit card transactions at 5%, and presumably something like this ratio would apply to digital money.

I believe that the brand identity and marketing aspects of such a service also favor a new approach rather than adopting an existing credit card or network service. The notion of having a universal account for on line information is novel enough that I do not think that it is a requirement to associate it with Visa, MasterCard or the phone bill. In fact, there are a whole slew of reasons to treat this as something new - it is the "card" you use with your computer (in the early phases) or with your smart phone or intelligent FAX. This is different enough from other services that I believe that establishing a separate brand identity would be quite straightforward.

With sufficient effort, any major player who handles financial services could attempt to build the right kind of services by themselves, but in every case it pretty much amounts to starting from scratch. I believe that we could make a compelling argument to such a company that doing it with Microsoft would offer enough technical and business advantages that we could cut ourselves in for a reasonably large piece of the business.

If we did, it would be very lucrative. Suppose that the service charged 3% transaction cost, and we got a third of it, or 1% of the revenue stream. This is comparable to the percentage of system price that we get for system software. The rule of thumb discussed above is that the amount of money spent on transactions is usually *many* times the node price. Once a service like this is established we could expect its revenue to also be *many* times the system software revenue.

In fact, one can view this as a "systems-like" cut of the transactions and billing business. Like system software, *the primary value is being delivered by somebody else* - the server operators which are selling information, physical products or services via the wide area information system.

Here are some of the attributes of the proposed service.

- Strong joint venture partner. We would want to work with a company which could provide both expertise and direct value added. Citicorp is the obvious example -

they understand the credit card business, and they can use Citibank to integrate the system with banking in a direct way.

- **Multi-platform.** We would aim to make Windows the flagship, but the general strategy would be to also support other PC platforms and a wide variety of PDAs and consumer devices.
- **Smart card or other physical security gadget.** We need some way to make this secure even on very insecure systems. A smart card or some security module (perhaps built into new modems?) would be a good idea.
- **On line (credit card like) and off line (digital money, coupons) are both supported.** This is important for providing a range of different service levels and transaction costs.
- **Tickets, time stamping and other digital certificate services would also be supported.** This includes being a digital notary public, providing unforgable audit trails and a variety of other variations on the basic theme. We could manage everything from frequent flyer miles to concert tickets.

Note that the goal of the system is not to be the only way that people pay for on line services, but we would attempt to be one of the early standard methods.

In the long run, one could imagine a system like this growing at the expense of traditional credit cards. Once smart telephones or other end nodes are commonplace, it would be easy to use the same system to start eating into traditional credit cards and cash machine cards. Some of these ideas have been discussed in previous memos about Wallet PC.

One obvious question is why Microsoft? How can we get ourselves into this revenue stream. There are a variety of points that I think we would use as leverage:

- **We provide the necessary technology.** It would be very difficult for a company like Citicorp to do this themselves, or even to find any other single partner that could span as many aspects of the technology as we could.
- **Can bundle the node in every copy of Windows.** One of the real goals is making this ubiquitous, and Windows gives us an opportunity to do exactly this.
- **Bundle the node end in every MS based PDA, EFAX and consumer device.** We can also put the necessary hooks into every relevant consumer device.
- **Can evangelize PC OEMs, modem manufacturers and others.** We are in a very good position to get people to build in modems, security gadgets and various other things which are necessary to enable the market.
- **We control RIP and help create the market.** Since our plan is to control the fundamental standard on which the wide area service is based, we are in a good position to influence the billing system.
- **We write and market the first real server apps.** The server software kit would include an account which is automatically registered with the billing service. We would make this particular aspect of the server open so that other server ISVs could also provide instant account.
- **An electronic software distribution system could use the transaction service.** The system described above is ideal for doing electronic software distribution. The server would either hold the app for direct download, or it could just hold a key used for unlocking a binary on CD ROM or some other means. We, and companies in our industry who follow our lead, could be the largest customers in the early

years because we push hundreds of millions of dollars of product through the system.

This combination may not be sufficient, but it is certainly enough for me to convince myself! I think that we should contact prospective partners right away.

Smart Card / Digital Billing Accessory

If we enter the transaction business as discussed above, there is a very natural market for an accessory product to both the node machines or the servers. This consists of hardware and software add ons which are either necessary to use the system, or which enhance it. Since the details of what we would do are not settled, it is hard to identify exactly what the content is, but I am pretty certain that some *accessory* is possible. Here are some possibilities:

- Modem. We should not get in the general modem business, but having a cheap modem which works with the system and is perhaps integrated into other hardware might be a very worthwhile thing to offer.
- Physical security device. Although the system would work without a smart card or other security hardware, it is likely that one would be needed in many situations, especially if the physical securing the machine is not sufficient. A smart card would also allow a number of other features, such as carrying your purchasing ability from machine to machine, keeping different accounts on different cards etc. The card would also be useful for other computer security uses (automated login etc).
- Socket for security device. Most systems would allow a fairly large number of cards - just as people can have multiple checking accounts or multiple sets of keys. A standard accessory kit would come with the "socket" which connects the card to the PC and one or two cards. Extra cards would be available separately.
- Digital coupons. As a promotional gimmick, we would bundle digital coupons which could be redeemed at various on line services for introductory subscriptions.

The exact definition of the accessory notwithstanding, the important business aspect is that we should have one for both the Windows based node and for the server. In the case of the Windows node, the basic software would be bundled free so the accessory would become our primary revenue generator at the node end. In the case of the server the accessory lets us capture extra revenue.

On Line Information Applications

The next obvious thing area to approach is on line applications, many (but not all) of which are primarily information based. If the transaction and billing service is the "systems-like" approach to on-line fees, then this is the "applications-like" side.

Note that there is an interesting relationship between these applications and multimedia information apps. The delivery technology is very different - the on-line version uses the telephone network, or some variant such as ISDN, cable TV etc to send digital material to a remote PC or node. This is almost always at a much lower bandwidth than we can get from a CD ROM, and as a result many of the cool things which are possible with multimedia cannot be done here. In a long enough time frame we will all have digital fiber connections so this limitation will go away, but in the meantime we have to plan around it.

A key part of any on-line application strategy is that we build in some synergy with off line apps, including CD ROM. Many of the relationships we build with publishers or content providers can have the dual purpose of serving both markets. CD ROM based titles can provide companion products to on line sources and visa versa. I won't dwell on this topic further - the key point is that the on line information business is *very* different from CD ROM, but with the appropriate strategy we can get some synergy between the two.

Having a strong on line application business is important for several strategic reasons:

- **Reinforces our system strategy.** Excel and Word for Windows have been excellent stimuli for the Windows market - they provided good applications for Windows earlier than other major ISVs, they gave other ISVs some explicit role models for what a good Windows app should do, and finally gave people an incentive to get off the dime and start being serious about Windows before we took the whole market. There is obviously a careful balance between reinforcing the strategy versus being so vigorous that third party ISVs are scared off. We have managed this balance in the past, and will have to do so again in this market.
- **Allows us to participate in other platforms.** Information apps, like our other applications can be an important way for us to address platforms other than our own. This too is an interesting balance - we need to take as much advantage as possible of our own systems and their unique features, yet at the same time if there is a high volume platform on which we do not have system software, applications are our only way to be involved. Lowest common denominator solutions usually are not viable on any platform.

Another principle from our experience with PC apps is that the best areas to approach are horizontal categories with broad appeal. The purpose of this memo is to talk about the business model, not product ideas, but here are some examples anyway:

- **Electronic software distribution.** Software is a very lucrative form of digital information, and has the great benefit that it is an existing proven market in which we are major players. It is not as exotic as some of the consumer oriented information streams, but the reality is that it already exists. An aggressive program for electronic distribution of our products is a great horizontal information application and also improves our base business. It should be one of the key markets for bootstrapping the system.
- **Service and support and customer information.** Another prosaic area is to supply as much as possible of our PSS services through on line servers, and to establish various forms of service, support and product information forums. The success of the Compuserve forums shows that this is feasible, and there are many ways to improve it. RIP will improve the quality of the interaction. Remote diagnostics and control of a customer's machine by a technician allow service and support to be much more effective, and to have various fee structures - free advice, but some fee for remote maintenance.
- **News.** General news, sports, weather, business news and other event reporting are certainly going to be an important horizontal application. "News" really stands for time critical information of broad interest. The trick for us is to figure out how we can carve out a chunk of this for ourselves. We do not have skills in gathering and compiling time critical information in real time, so it is clear that we have to either spend a lot of effort to acquire the right people or work with a partner that is in that business.
- **Travel information.** This is basically like news with a slower time constant - most travel information only changes with a monthly or annual frequency. The key thing

about travel is that it is an expensive and frustrating thing to do, especially if you lack the proper information, so people are willing to pay more for it. Once again the key issue is how we can be involved given that we are not familiar with the details of the information.

The last two examples illustrate a general point which is that we are not authors, editors, reporters, cartographers or specialists in other forms of information creation. The first two example above just about exhaust the areas in which Microsoft can reasonably expect to run the entire gamut from authoring to packaging and delivery. The question this poses is how can we leverage our existing skills to earn a slice of the business where we (at least in our current organization) will do not have the skills to do the authoring and editing.

One answer is to just roll up our sleeves (and take out our checkbook) and build a new group which does do the authoring and creation. In the last 10 years CNN has gone from a small operation to become the most important news gathering and distribution company on earth, so it certainly can be done. One can even argue that in the long run this is bound to happen because the new electronic medium will be sufficiently different that it cannot accept "hand me down" information from an old sources any more than TV news consists of reading newspaper articles.

The problem with the all-from-scratch approach is that it is very expensive and consumes a lot of our bandwidth. We could address many more information products and leverage our own efforts if we could find partners instead.

Starting a new information organization requires a fine sense of both timing and patience. It is hard to predict with certainty when the market is mature enough to launch such a venture - if you go too early you will not only burn lots of money, but will also lose good people because it is frustrating to be "ahead of your time". The time slogging through the trenches is not even good for much, because it is hard to learn without real customer feedback, real delivery systems etc. If you wait too long to start a new service you may find that somebody else who reuses existing information might get in ahead of you. An incremental product might take a long time to be as good as the "from scratch" system, but this may not matter if the incremental version establishes a dominant presence by virtue of being there first and getting the benefit of early customer feedback.

I think that the best approach for us at this stage is to go the partnership route, and to focus on the areas where our natural skills are poised to the best advantage. This means:

- Create proprietary tools. The big hurdle for any information producer is to get the information in a rich digital format meeting the constraints of time and cost. Software tools which are designed for the production of large scale time critical information will be at the heart of any successful information service. The existing people in the news, travel or publishing business will not be able to develop these themselves - *we can*. When we do, I believe that we can use them to leverage ourselves into a partnership role.
- Develop expertise in using the tools. The production process is just as much a black art as the tools, and it is another area in which we should invest. Since much of the process and technology is brand new and without much precedent, there is nobody more suited than we are, especially if we are using our own tools!
- Use a combination of this tool technology, our market presence and bravado to get good partnerships. I think that it is necessary to focus on all three areas in order to be successful. As Tom Corrdry has pointed out, market presence and bravado are enough to get us luncheons with nearly anybody, and we have been on

that circuit for quite some time. We need to focus on some hard core technical benefits and a sound strategy to actually land some of these ethereal deals and then go on to make them successful.

- Get in early. Magic tools, market presense and so forth are great bootstraps but ultimately our presense in the market will, if done correctly, get us a an enduring franchise which will last long after the bootstap phase. This is particularly true if we can be there from the onset. Note that the downside to this is that we may have one or more disappointments where the market does not take off the way we would like and we have to start again. The only reliable way to be there first is to make your move at a stage when there is still some risk.

This strategy is similar to the one advocated in my memo on *Internal Program Development Tools*. Good tools can be excellent products and important ways to gather third party support, but *really* good tools can be put to better use internally as a proprietary advantage. The experience with P-code in Microsoft applications is a good example - this technology made a *decisive* impact on establishing our Macintosh apps, and I believe that some analogous tool could help bootstrap us in information applications.

We obviously will need to have authoring tools which are products, and I am not suggesting that we cripple them. Nevertheless, there is a large difference between making a PC-centric tool for small servers and a set of industrial strength tools which could be used for a national news service. Small scale tools are great retail products, and, at least in the early days large scale tools will get us a healthy ongoing percentage interest in new ventures.

The Nintendo Model

Nintendo (and others before them) is famous for its policy of making the hardware very cheap and then charging ISVs enormous royalties (by our standards) and subjecting them to a great deal of scrutiny and control (only approved games can be sold). This is clearly a winning strategy for them at this point in time, but I think that it is the kiss of death looking forward into the future. I think that it would be a disaster for us to adopt.

The reason that Nintendo got away with this is that they were able to achieve critical mass based on their own games, and added ISVs as a second stage. I do not think that this is a very likely scenario for us, nor do I see it as being worth betting on. When it comes to a diverse market with many niches I believe that an open system will always dominate.

Compuserve and Prodigy have each attempted to be something more like Nintendo - toll takers who offer a selected number of services. They have been less hard core than Nintendo in terms of rates, but that is not the real issue. The fact that they have not realized is that closed worlds like these can never have the breadth or depth that an armada of independent services can have.

The scalable server strategy is based on the belief that thousands of server operators will find ways to make money offering on line information and in the process of making a buck for themselves become missionaries for our systems, applications, transaction services etc. As they do this, they fill in gaps in the range of services available and create a sense of ubiquity which will draw many people into the on line phenomenon who would never do it for a small fixed set of services. Owning a FAX machine five years ago wasn't much fun unless you knew somebody who had one (much like having a video conference room today). These days, you can count on nearly any business to have them and it is only a matter of time when they will be as pervasive as telephones. We need the same thing to happen for both servers and nodes (smart phones, PCs etc.).

This is the antithesis of the Nintendo model. It would be unworkable to try to charge the hordes of servers anything more than a one time fee for their server application, and we certainly couldn't have the "artistic" control that Nintendo exerts. Even if you found a way to do it, it would be missing the point - the marketplace should fill in the niches not some sort of central planning.

The price of opening the floodgates to all comers is that large players who want to do major applications with broad appeal don't have to pay us any more than the corner deli, or a guy who trades rare stamps from his PC. The solution to this is to create things like the transaction and billing service, and wrangle our way into some of the horizontal applications by virtue of tools technology.