From: Craig Mundie
Sent: Sunday, March 04, 2001 7:09 PM
To: Bill Gates
Cc: Eric Rudder; Christine Turner
Subject: RE: Executive retreat

Importance: High

Bill - I am in town until a week from Tuesday, when I have to leave for China. As you may recall, we agreed that I was going to skip the retreat to go meet with the Chinese premier, and that appears to be happening, with meeting in China scheduled for the 15th and 16th. However, I am willing to clear a bunch of stuff from my calendar between now and then and prepare the three hours of the future stuff for you I will not be able to be there to present it, but I am sure that I can get some stuff together as pre-reading and for you to present at the retreat. Is that OK?

Below you talk about a memo being edited, and I assume you are talking about the one that I sent you a few weeks back. Peter and I did finish it and I sent it to the SLT and selected others. I didn't send to the exec staff, but we could. I am also building a cool future demo to use in my WinHEC speech which is a "photo scenario" that does all four of the levels that I discussed in that memo: PC+Peripherals, PC + LAN connected devices with WTS, PC + Devices operating Peer to Peer across the WAN, and access to premium services. Since we don't have any shared space/peer to peer tools yet, part of this demo is actually built on Groove! It also uses a bunch of research stuff for searching and organizing your photo collection based on image classification. The UI is built using the data mountain metaphor so as to give the appearance of a "content addressable storage system" (aka Yukon?) with new user graphical interface abstractions for the user to find, organize, manipulate information in interesting ways There are no "folder trees".

Maybe we could send out the memo that Peter and I did (supplemented by another memo from you that touches on other topics below), and then show this demo to make it more real... I think we could get this ready in time for the retreat. Enough of this is working now that if you came over to my office/lab (it isn't very portable just yet) we could show you that and we could discuss what else you want me to pull together from the list below... Also, Pierredv and Suzew and I have been brainstorming about "eWork" in preparation for engaging with Raikes and Sinofsky about building an Office of the Future and helping them think about an expanded concept set/mission for Business Productivity Group. I might be able to write up some of this with Pierre and Peter Haynes this week if you wanted to toss that into the mix, but it would be much more speculative/brainstormy but it might be great to get people to work on a report against ideas for a next generation of BPG applications.

I have some time for 90 minutes tomorrow morning starting at 9:30. I will move things around to do this work and accommodate your schedule to the extent it is possible. Calendar is currently pretty full this week, but not traveling this week, and I could resched some of these for a couple weeks from now...

I have attached the final version of the "Internet V3 + Experiences" memo here if you want to review, or if you want Ericr to think about this as part of a pre-reading kit for the retreat.

Realizing Internet
V3 Final 0...

Also, Toutonghi had a pretty good whitepaper that he wrote about his views of the eHome stuff. Tim Bucher has some stuff too. We could move the Galaxy demo up to the retreat. I have a early version of it running in my conference room now that we used to brief Allchin when I was trying to get him to understand the eHome stuff better. There are some other Connected Home demos that you could probably get up there pretty easily too. There are also some other demos being prepped for WinHEC that could add to this. I could give you my "speak to the iPAQ" demo, where I give it voice commands and tell it to play music which then gets a PC to stream music to a Euro Sony Stereo system. You can also fill out some forms using the MiPAD technology. I also show streaming video over the 802.11b network, with the iPAQ acting as a terminal SERVER so that people can see the video presented on a large projector by a standard PC This touches on a bunch of things related to new UI. eHome, devices in business applications, streaming media, wireless networks, UPNP all in one demo...

I love showing the Horvitz stuff, and I also have some PPT slides of key aspects of his Priorities system so that you can

Plaintiff's Exhibit
6949
Comes V. Microsoft

talk about this work without having to try to show it live.

I also have a demo that I do where someone can sit in the audience and make a videophone call over the wireless LAN to my iPAQ which I have in my hand on the stage and which is being projected using the reverse terminal server stuff. Here you get the VOIP stuff, and the highly compressed video stuff from MSR china. It is currently set up where I see the caller, and I can talk back to them but I don't have a video camera on the iPAQ so they only hear me and can't see me. You could do this demo with two SONY laptops with integrated cameras and get bidi-video, but it actually seems cooler to demo this to the iPAQ. This portends a future for Stinger which includes rendering video phone calls...

Pierredv's group just finished a live demo of "context", which you could access over the internet from the retreat. In it you can click on a link and select someone and it will tell you what it knows about them. It tell you where they are based on the wireless network triangulation on campus, it tells you where they might be available based on the IM status, it gives you contact information from the corporate AD data base. It is pretty cool. This is accessed through a browser, and could probably be rigged up to demo future capabilities of pocket PC/Phones and also the idea of a screen call of sorts.

At the recent CIO summit I gave a live demo of data warehousing and data visualization showing new features from Heckerman's research in both SQL 2000 and also a prototype of new tools he will be putting into Yukon. Coupled to discussion of XML, shows new cool analysis that can be done. Used with MSNBC data set it taught the MSN web team that actual use of the site was totally different than their design assumptions, and they have changed now... This runs on a laptop and has cool graphics...

I know that Will Poole recently gave a cool 720P24 Video demo running off a PC. This is progressive scan HDTV as we have been evangelizing it for a few years now. It runs from a relatively low bit-rate streaming video source. This could be set up, with some effort, in the retreat, but it would be cool to keep it running and show people the quality of the video that the PC servers will be able to provide/render. Also, there are a number of cool demos in research here in Redmond and from China in the area of MPEG4 object compositing, automatic re-formatting of web pages for small screens, video editing,

So, this set of demos that I already have, and give variously in my current set of speeches, covers the following from you list below:

1. PC to iPAQ video phone calls with the possibility of talking about this as a Screen Call. Another demo is built on Groove and shows P2P.
2. None
3. None
4. Horvitz stuff or slides. Pierredv's context services demo.
5. None
6. Lots of this stuff. MiPad is new Dr. Who, which I use for the speech demo on the iPAQ.
7. SQL/YUKON/XML warehousing and realtime data viz stuff.
8. Ehome vision – Memo and four-level photo experience demo with smart picture frames and PC and P2P operation...
9. None
10. Other whizzy demos from MSR...

So, this demo set, which I have almost completely set up to be portable (the new stuff was to be portable by WinHEC, but we would try accelerate it for the retreat) covers a pretty big chunk of the stuff you talked about in your list. I could have Ron Day who travels with me and sets this all up, go up and support you doing this complete set of demos at the retreat. I could train you personally on how I do these demos and what the story I tell about them is. I am here both days next weekend too, and would be happy to meet you over the weekend to train you up on this stuff. I think it is so much better when I (or you, in this case) do the demos rather than have other people do them. We are saying it has to be easy, and so I work hard to make it look easy! ☺

I could ask Suze Woolf who works with my on my speeches and has the little group that builds the fancier demos to help pull this together and support you up at the retreat if you want, too.

-----Original Message-----
From: Bill Gates

Sent: Sunday, March 04, 2001 4:37 PM
To: Craig Mundie
Cc: Eric Rudder; Christine Turner
Subject: FW: Executive retreat

Craig – How does your calendar look between now and the exec retreat?

Unfortunately I am at a Retreat this Tuesday, and New York Thursday and Friday and in Denver on Monday in the morning.

However if you were willing and able to take the lead on the "3 hours of the future" stuff I discuss below that would be great

If so we should squeeze 30 minutes in between now and when I leave for New York to discuss this.

Eric – if Craig does this then the thing I would want to focus on with you is the idea of issuing the memo

-----Original Message-----
From: **Bill Gates**
Sent: Sunday, March 04, 2001 11:34 AM
To: Eric Rudder; Craig Mundie
Cc: Liz King; Senior Leadership Team
Subject: Executive retreat

Among all the other things going on we have the exec retreat coming in less than 2 weeks.

Liz King came by last week and told me that they are thinking I will have 6 hours total set aside for what I want to do. They haven't gotten input from Steve yet so this could change though. Are Des invited to the exec retreat?

As you saw from my calendar I don't have much free time between now and then to have pre-meetings.

There are 3 hours on Thursday afternoon and 3 hours Friday morning.

I want people to understand my excitement about a number of new areas.

Some of these things it hard to know what to show either because they are technical or not ready yet. I love Xbox but since I think we can fill up the time without it I don't think we have to show that.

I am tempted to have someone "present" where we are going with the thing we demo so people don't just see the near term.

1. Issue some form of my Memo to the group before they come. I have NOT looked at the editing work you and Peter did but I will get time to do that – maybe not until next weekend sometime. I think about half of the execstaff has seen the memo in some form.
2. Ask some people to demo and discuss cool stuff on Thursday. This would mean me not speaking much at all during that time
3. I talk for say 90 minutes on Friday and then we have tables of people who discuss what I said and which things are important and how we achieve them and then report back. The tables get to talk for say 45 minutes and we have 45 minutes for report back. If there are 9 tables of 9 people then each table gets 5 minutes of report back.

The key thing is to explain the breakthrough things that are exciting and involve cross group scenarios.

Here is the list of things I consider in that category:

11. Real time/Screen Call/Apps to facilitate. The idea here is that the majority of pure calls today will use Pc screen-screen with software to help with the call. Groove touches on this some. My screen call memo does some. Not sure what we would show here – the next Messenger client? The Netdocs stuff we just saw? Just some slides? This is an important one and I would love someone on the exec staff to be a champion of it. One aspect of this is real time for support scenarios – we could get someone to show that.
12. Reading/annotations/notes/Tablet. Alex is not a VP so this would have to be Dick or Jeff unless we invite her specially. When I say tablet I actually mean a lot more than tablet. I mean READING. I mean ANNOTATING. I mean NOTETAKING. Bill Hill gets reading. Darryl gets annotating. Bert Keely gets note taking. I do want people

to see the device and understand the status of where we are.

13. Completing the feedback loop with customers and Magic software distribution. This is sort of about curing sins. However it is so central to our future. Show IM update? Mars? Discuss what we have learned? Windows Update/PC health plans? Dr.Watson experience? Managed PC?
14. Information agent, Real Collab and Workflow based on XML. Biztalk is just a small piece of this. I don't know our CRM stuff. Could show some of the net .NET demos here. Could show Horvitz stuff.
15. The value of Integrated storage. How the UI can be simple and effective with a minimal number of verbs. How the programming experience is better. How bringing caching/queuing in helps. How the rich data model helps.
16. PDA futures. Some of the phone/PDA stuff they are doing is nice. The demo won't show the future just the current piece. Voice/Dr.Who? Discuss scenarios that we can do uniquely.
17. XML stuff at the end user level. Data Warehouse? Maximal? (not sure what happened on that)
18. Ehome vision stuff.
19. Programming environment advances. Probably I should just talk about this one.
20. Other Pc Excitement stuff? Do I miss something we are doing with the PC in the above?

I think this is a great chance to get the exec staff enthused about these advances which will expand our profitable businesses and keep them healthy as well as tell people inside and outside the company that our focus on great software is leading to some amazing breakthroughs.

Realizing Internet V3 **Craig Mundie / Peter Haynes** **January 2001**

1. Context

The evolution of both telecommunications and computing has been marked by periodic shifts in the primary location of network intelligence. In telecommunications, this phenomenon has been well documented by David Isenberg, notably in a paper that is worth reading for its parallels with developments in the computer industry.

The evolution of intelligence in the computer industry, however, has been much more complex and erratic than that of telecommunications The mainframe computer placed intelligence (i.e., processing power plus storage plus software) squarely at the center of the network, with read/write functionality but no processing ability available to non-technical users at remote terminals. In the early 1980s, the standalone PC reversed this, putting all the computing power in the hands of users. Then the advent of PC-based servers redistributed that power once again, shifting the balance towards IT administrators but still leaving immense power in the hands of non-technical users. The microprocessors and software in today's typical PC are considerably more powerful and sophisticated than mainframes and workstations of the early 1990s.

The last few years have seen parts of the industry move back towards the centralization of computing power/intelligence. In part, this has been a consequence of the marketing strategies of companies such as Sun and Oracle, which have tried to champion the concept of the "thin client," albeit without significant commercial success.

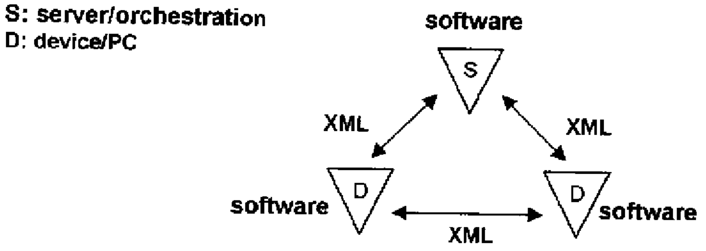
But the main technological driver behind the rebirth of centralization has been the Internet, and the growth of the World Wide Web. In several respects, the Web mirrors the old mainframe model. Information is locked away in centralized databases. Users must rely on the Web server to perform every operation, just like the old timesharing model. Web sites are isolated islands, unable to communicate with each other on a user's behalf in any meaningful way. The Web simply serves up individual pages to individual users—pages that mostly present HTML "pictures" of data, but not the underlying data itself. And the browser is a glorified read-only dumb terminal—you can browse information, but it is difficult to edit, analyze or manipulate (i.e., all the things knowledge workers need to do). Today's Web is a long way from the "intercreative space" envisioned by Tim Berners-Lee, whose pioneering work lead to its creation. It is also a long way from meeting the needs of both consumers and knowledge workers.

As a result, we are likely to see a significant move away from this centralization of computing power during the coming decade. Today's Internet runs counter to computing's natural centrifugal force, which constantly creates more power/intelligence at the edge. Already, microprocessors are diffusing into every kind of device and appliance, enabling complex software to run locally even on small devices. To understand why this decentralization of computing intelligence is likely to accelerate, consider the probable evolution of the Internet during the next few years.

The Internet has now been through two iterations: the first version (V1) was largely an academic and government incarnation that was focused on interconnection and the movement of files. Today's Internet (V2) is largely an environment where Web sites publish, and people consume, pages. To take the Internet to the next level (V3), we need to drive a transformation that increasingly will see today's HTML-based presentation of data augmented by programmable XML-based data.

XML provides a means of separating actual data from the presentational view of that data, and is the key to creating Internet V3. XML "unlocks" data so that it can be organized, programmed, edited and exchanged with other sites, applications and devices. In effect, it transforms the data behind every Web page into a programmable mini-database, and enables different sites to share data without having to use the same computer language or software application. Individual sites can collaborate to provide a variety of Web-based services that are able to interact intelligently. And information can move easily from one device to another without the need for today's separate applications, with their widely varying interfaces, functionality and (in)compatibility

Internet V3 will see another significant paradigm shift The Web as a paradigm has yet to mature to the point where computers (rather than individuals) consume pages—where the primary traffic is *computer to computer*, not computer to user. A total redefinition of how many businesses operate, and of how devices work together, will come from creating the computer-to-computer generation of the Web to augment the computer-to-user generation. The *lingua franca* of XML will enable these orchestrated "software-to-software" or "machine-to-machine" interactions between Web sites, and make possible an almost unlimited range of customized Web services. So we end up with a world that looks something like this:



Microsoft's .NET strategy will both leverage and drive this paradigm shift. Built on the standard integration fabric of XML and Internet protocols, the.NET platform is a revolutionary model for developing an advanced new generation of software. Previously, programming models have focused on a single system, even attempting to mask interactions with other systems to look like local interactions. .NET is explicitly designed to enable the integration or orchestration of any group of applications and resources on the Internet into a single solution. Today, this type of integration is both complex and costly. If we execute successfully, .NET will make it intrinsic to all software development

The loosely coupled XML-based .NET programming model introduces the concept of creating XML-based Web Services. Whereas today's Web sites are hand-crafted and don't work with other sites without significant additional development, the .NET programming model provides an intrinsic mechanism to build any Web site or service so that it will federate and collaborate seamlessly with any others. Just as the introduction of interchangeable components accelerated the industrial revolution, .NET promises to hasten the development of Internet V3.

If the paradigm shift described above is inevitable, as we believe, the consequences for today's online and ecommerce business models are profound. In particular, a shift from publishing Web sites/pages that are consumed by an individual user to "publishing" Web services that are programmatically consumed by an application, means that the notion of an online advertising model (whether targeted or not) goes into decline—or is at least rebalanced vis-à-vis alternative financial models such as subscription or transaction-based payments.

By definition, the advent of Web services means that the need for power/intelligence at the periphery increases. Just as today's modest Web services (such as Napster) requires both local and central processing power—i.e., for personalization and orchestration, respectively—tomorrow's Web services will require a balance of centralized and distributed intelligence. This is partly a result of the fractal, loosely coupled nature of the next generation Web: everything is recursively reconstructed as a Web, regardless of the scale at which you examine it. If you zoom in from Web scale, to enterprise, to cluster, and then ultimately down to individual PCs and devices, what you'll see at each level is a collection of tiny autonomous sub-systems, each part of an interconnected Web... that is in turn part of a wider Web.

A fractalized construction pushes computing power and intelligence to the periphery of a network. Latency and/or jitter ultimately limit the degree of centralization in a fractal Web, particularly as computer clock speeds continue to increase. Latency and computational intensity are why you probably wouldn't, for example, perform vision processing in real-time back at the server.

The infinitely fractal nature of such networks is also a reason why simple peer-to-peer applications will be of limited use in this world. Diversity and scale are the two key hallmarks of fractal network construction, and beyond a fairly low level of (a) device diversity, (b) participants and (c) complexity of association, you need some degree of orchestration (either in the network or in the cloud). This is one reason why applications such as Groove will, in a world of seamlessly federating Internet resources, likely achieve only limited penetration (although whether Groove is pure peer-to-peer or marginally orchestrated peer-to-peer is open to debate).

2. Taxonomy

To better understand the role and potential of Microsoft in this new world, it helps to categorize the technology by four levels of functionality/connectivity:

1) The device (and its ability to connect *directly* with a PC);

2) The local area network (and its ability to link PCs and devices remotely *within* the home/ or office);

3) The Internet (i.e., the wide area network that enables PCs and devices to link *beyond* the home or office, and also facilitates peer-to-peer connectivity);

4) Premium services (the provision, administration and orchestration of resources to leverage the functionality/connectivity of (1) through (3)). A primal form of premium service may in fact be the sale, support and maintenance of the set of client applications that provide the user experiences at levels (1) through (4).

Note that this basic four-level taxonomy maps to all the *experiences* Microsoft is focusing on: photos, music, various forms of video, communication (telephony, email, instant messaging), home control, and access to information (whether published by my refrigerator or by an entity in the cloud). We need an architecture that enables smooth and modular upgrading from a single device to a LAN to a peer-to-peer or orchestrated peer-to-peer environment in the WAN and ultimately to the incorporation of premium services in that environment. Today, we have people in the PC group who principally think about level (1), and people at MSN who principally think about level (4). But what the customer really cares about is applications and related services that enable the graceful and seamless migration from—and integration of—levels (1) *through* (4). It follows that levels (2) and (3) represent a new demand on the aggregate platform—*and it is the linking of all four levels and experiences that will give us a sustainable competitive advantage.*

3. Understanding Experiences

To win, Microsoft must offer a compelling scenario direct to the consumer at retail—a solution that provides consumers with a compelling benefit from linking the PCs that are already in their lives to a range of new devices. We need to develop suites of client-side code that can work across numerous connected devices—sophisticated software that will bootstrap new and compelling experiences across all four levels.

In the past, Microsoft has always won by being neutral to both infrastructure and content. In the multi-dimensional Internet V3 world we envision, this becomes even more important—but if we are to win, that neutrality needs to be leveraged across all four levels. At present, we have too few application and service assets at levels (2) and (3). But there are huge opportunities both to create them, and to encourage others to do the same.

We can use a simple telephony-based scenario to illustrate this. At present, there are a large number of constraints on users in the telephone world. There's always a gatekeeper between you and your ability to do what you want to. First, your phone has to have (e.g.) proprietary Nokia and phone.com software in it. Second, it all has to interface with the proprietary software/infrastructure that the carrier has provisioned, and it if doesn't, it won't function. If you want additional services, you have to wait until the phone manufacturer and carrier offers them. You are wholly tied in to that closed system, and you have no control.

Now assume a new kind of telephone—the Microsoft CoolPhone service. It has a color screen, so you can carry your kids' photos on it. It has a lot of storage and a socket for headphones, so you can listen to music on it while you're on the move. And it has a Bluetooth chip, enabling short-range wireless connectivity at a megabit per second, or about ten times as fast as a serial cable or infrared link. So if you are at home the phone affiliates with your home server and PC. When you are in the office it affiliates with your work PC and LAN There won't be any perceivable difference between making a regular phone call and a peer-to-peer voice-over-IP call, because the phone you're calling and the phone you're on are both accessible on the Internet. And when you are out of Bluetooth range, the phone reverts to being a cellular or land-line wireless phone, working over your carrier's network.

The key advantage to the user here is that they don't need to wait for the phone carrier to provision these services You get everything through your PC regardless of who your carrier is, which makes this a very appealing premium product/service package for consumers. This is a huge value proposition for Microsoft, and gives us a compelling retail offering. The phone is participating in a Microsoft photo experience and a Microsoft music experience, and so on, and consumers don't have to wait for, or get permission from, their carrier. Consumers can even decide what kind of phone service they want to use (voice over IP or traditional) without making any changes to their basic telephone service. And for other services, instead of having to make a phone call to download bits over the carrier's network, you can just suck up the bits you want from your nearest PC or server. This puts incredible power into the hands of the consumer, and it provides Microsoft with an attractive premium service—*and some valuable "sticky" assets" at levels (1), (2) and (3) of the taxonomy above.*

The core strategy here is *leading with the experience, and letting this pull the user through the aggregate platform*. If you create demand for the experience (in this case, the experience offered by the Microsoft CoolPhone), then the suppliers of both basic (e.g., telco carriers) and premium (e.g , AOL) services will be compelled by customer pressure to ensure that their infrastructures deliver that experience more effectively to consumers. This should be familiar territory for Microsoft. The key driving force behind the widespread popularity and adoption of Windows was the experience offered by the Office applications suite.

By leading with the experience, you win a larger share of the value proposition. Users don't have to be affiliated with AOL or MSN, or use Sprint's or AT&T's network to benefit from the experiences Microsoft is offering; those experiences are agnostic with regard to every aspect of infrastructure, but at the same time sticky with regard to Microsoft The technology will ultimately create multiple alternatives at the pipe level, at the infrastructure level, and at the device level In a world where bits are affiliated with numerous service providers (AOL, MSN), transmitted across several different networks (device-to-PC-to-device, LAN, Internet) via various media (wired, wireless, cable, etc), to a wide range of hardware (PCs. devices, appliances), the *experience* is increasingly defined by the abilities of the aggregate OS in the cloud. In other words, .NET.

We need to be aggressive in creating experiences and value that are independent of anything other than basic Internet access. By doing this at retail, aiming directly at the consumer, we create "sticky" assets—assets that are both innovative and *persistent*. If the experiences Microsoft creates touch multiple devices—so consumers have a variety of devices that are already tied to, e.g., their music experience—then when given the choice between buying a phone that ties into that music experience and one that doesn't, they will choose the one that does. Even if it comes from Samsung instead of Nokia, or is offered by AT&T instead of Sprint. Because it's easier to change one device or service rather than forgo the music experience that works with multiple devices. In several respects, this strategy is similar to MCI's "Friends & Family" scheme. We need to create Microsoft Friends & Family experiences

4. Creating Experiences

Today, we sometimes appear to view user experiences either as (a) a feature of the OS, or (b) generic in the Internet sense (MSN and other Microsoft online assets notwithstanding). But in the new world we are describing, the platform is actually a union of (a) intelligent devices, (b) OS's for those devices, (c) tools to program those devices, (d) a set of network-based services, and (e) the programming model that invokes those services. That aggregate platform yields experiences that consumers can subscribe to and access by whatever means is appropriate. *And, in turn, those experiences are applications plus their related services.*

If this logic holds, then Microsoft should be selling infrastructure to people who build the service end of the experience, and both platform *and* applications to end users. If we write those applications (and create an evangelism program to ensure that third-parties are writing related applications), then we will have a unique selling proposition to take direct to the user. We will also create experience-based integration between what users can do on a single device and what they can do in the network or cloud. If we don't write the code that runs in the clients, we won't create compelling experiences (with or without a service component). We need to create a new generation of killer apps for the workplace and a new generation of killer apps for consumers.

In a previous paper[1], we described such a suite of consumer-oriented applications and related services that we called Microsoft Life. In the loosely coupled, connected world we envision, this suite would be to the home what Office currently is to the enterprise. It would be capable of working on all the smart devices in the home, and would offer a framework for integrating many other devices and services (e.g., in the same way that HP makes printers with font support so you can print the documents made in Word). Each component would have sufficient similarity in their UIs that consumers could move seamlessly from one to the other. It would be the essential software that supports the consumer as an individual outside the enterprise and as a member of various communities, starting with the community we know as "family." And each component would share fundamentally the same architecture, and offer a similar and familiar interface.

We continue to believe that such a suite should be at the core of our "consumer experience" strategy. When we initially described Microsoft Life, we envisioned five core application/service components that would be available wherever you wanted them (e.g., on your PC, music player, pocket PC, eBook, Bluetooth phone, etc): *entertainment* (music, TV, video, photos, games, Web-based services); *information* (Web, print, broadcast and custom media); *communication* (the family mailbox—telephony, instant messaging, email, vmail); *convenience* (home automation, home shopping, health and safety services); and *productivity* (in addition to Office-derived applications such as Word and Money, a core application here would be calendaring). However,

[1] "Microsoft Proximity Namespace & Microsoft Life," October 1999

the components of an initial release of Microsoft Life could just as easily focus on more limited functions, e.g., photos, music, video, communication, home control, and information access).

We might also consider linking such a suite to a core media store—a place where you can put your books, music, photos, movies and so on, and which you can annotate, search, edit, stream, etc. Such a store would enable you to create "playlists" for each type of media without having to learn fundamentally new concepts for searching and organizing each type: like the other components of Microsoft Life, it would share the same fundamental architecture and interface. It would also offer users a personal "information agent" that would help you choose music, books, TV programs etc by learning from your own purchasing patterns.

5. Opportunities

The opportunities for both Microsoft and programmers in the world we envision are immense. Most of what are likely to be the most important and useful Internet applications have yet to be written, while the currently available experiences are still very basic. By writing the killer apps (and related services) that drive these experiences, Microsoft could (a) win a larger share of the value proposition; (b) create a set of sticky, persistent assets that drive both platform (.NET) adoption and profitability; and (c) help empower third-party programmers by driving the platform and enabling them to focus on areas where they can really add value. For NET to succeed in the long term, we must make this a priority.