Hypertext — Does It Reduce Cholesterol, Too?

Hypertext '89 Keynote Address
November 6, 1989
Pittsburgh, PA

Norman Meyrowitz

Institute for Research in Information and Scholarship (IRIS)
Brown University
155 George Street
Box 1946
Providence, RI 02912

IRIS Technical Report 89-9

Prelude

Frank Halasz

Hypertext '87 started with a speech by Andy van Dam from Brown University, who gave us a very nice history of the work that has gone on at Brown University and other places over the past twenty years. So it's very appropriate that this time we kick off with the next generation of Brown University research in hypertext, Norm Meyrowitz.

I first met Norm at CHI '85 in San Francisco or Boston, I can't remember. Randy Trigg and I had brought our new system called Notecards to demo. We lugged along all this big fancy Xerox Lisp equipment and had this really whizzy demo all set up to go on what I think was officially part of the demo series. About an hour after we started there was this crew from Brown University with their little IBM PC RT, probably half the equipment and half the cost of ours. * *I don't think they were officially on the demo schedule — they sort of had an illegal demo set up. They showed everything that we had, and more, on a cheaper platform, and boy was I jealous. Since that time, I've watched with great admiration as Norm and his crew from Brown have continued to develop Intermedia to be a really wonderful system, something that I think has been, for the last few years, the most interesting and complete hypertext system around.*

Now for the official introduction. Norm is the Associate Director of the Institute for Research in Information and Scholarship at Brown University and has worked at Brown and IRIS since he graduated from Brown in 1981. So with great pleasure, I introduce Norm Meyrowitz, who will talk about hypertext and its effects on our health.

Norm Meyrowitz

Before I start we should also give a hand to Frank as the Program Chair who spent a lot of time on the telephone listening to us on the program committee and other committees complain about this, that, and the other thing and managed to do it with great humor and put together a great program.

And finally before I begin, I'd like to say that this isn't really a keynote by me. It's a keynote by all the colleagues I work with at IRIS — this talk represents all of our work together.

So with that, let me start on the topic of "Hypertext — Does It Reduce Cholesterol, Too?"

* It was San Francisco, and it actually was a 512K Mac running a very early Intermedia prototype.

Hypertext —
Does It Reduce Cholesterol, Too?

Norman Meyrowitz

Institute for Research in Information and Scholarship (IRIS)
Brown University
155 George Street
Box 1946
Providence, RI 02912

Introduction

If you look back just two years ago to the Hypertext '87 conference, there were literally only two books available with any great mention of hypertext. These were Ted Nelson's Computer Lib/Dream Machines and Literary Machines. By 1989 there were a dozen books fundamentally about hypertext that I could gather from my office in under three seconds. And there are literally scores of others that purport to be about hypertext that are sprouting up like dandelions. One hypothesis is that all the books that were supposed to be about cold fusion were reprinted with new covers about hypertext.

So what's happening with the field? Part of what Frank said this conference should concentrate on is looking at the research that might inform us what the commercial offerings will be in the years hence. So I took the liberty to put together a couple of prototype slides of what one vision of the field might be one year out [Figure 1]...

Figure 1

three years out [Car w/ "Hypertext option no extra charge" sticker]....

and five years out [Dog w/ "Now with hypertext!" sticker].

I wonder and I worry that hypertext may be the oat bran of computing, that we may be so excited about perceived potential that we miss some of the real advantages and some of the real impact that hypermedia and hypertext could have amidst all of the hype. What I want to do today is look at the impact of hypertext and hypermedia and how it can survive and be something nutritious for us all.

What Is Hypertext, Really?

First we have to figure out what hypertext is. To some of us it's graphical programming à la HyperCard. To others it is outline processing à la More or Acta. To others it is multimedia presentations that you make to the board of directors containing a lot of dancing dollar signs and so forth. And to others it's simply glitz replacing substance: no longer do I have to think because now these pictures are moving and moving pictures look interesting.

One can glean, by reading the conference proceedings and tracking the press, that other people think that hypertext is fundamentally a tool for teaching writing. There are people who think that hypertext is a reference tool, a way to put encyclopedias or dictionaries online. Some are focusing on hypertext as a way to do argumentation, as a way to focus and connect the different parts of an argument. Some are looking at hypertext fundamentally as a backbone for group decision support, as a way to put together all the disparate decisions of an organization. Others are excited about hypertext for interactive fiction, as a mean for creating new types of novels. Others are looking at hypertext as a way to do annotations, commentary, and criticism. Still others see it as the library of the future, a grand vision where hypertext replaces the current public libraries. Others who are represented at the conference see hypertext as the physician's dream, where all the medical textbooks are online with immediate access to the information that is needed, or the attorney's dream where all case law, all briefs, and all citations are immediately accessible. And still others see hypertext as a funding mechanism now that their object-oriented programming grants have run out.

Hypothesis

So my hypothesis is that hypertext is all of that, but hypertext is more. Down deep, we all think and believe that hypertext is a vision that sometime soon there will be an infrastructure, national and international, that supports a network and community of knowledge linking together myriad types of information for an enormous variety of audiences. So when we speak of hypertext and hypermedia in terms of our dreams and our passions, we're talking about having information at our fingertips in all the natural ways. Hypertext may be too narrow a term — we're envisioning this exciting information environment of the future.

A Roadmap

To give you a roadmap of the talk, or a trail or a path, what we're going to do first is take a hypermedia tour through some of the Vannevar Bush papers —*As We May Think* and *Memex Revisited* —looking at what he said and trying to correlate what he was talking about with where technology is today, how far we've advanced, and where we yet have to go. Then we'll look at the Memex of the 1990s, at what a potential information environment might look like given the context of today's machines, marketplace, and where things are going. Finally, we'll look at some challenges in the field, what I think the major challenges are for all of us if hypermedia is to have the payoff that we all hope it will. So I'm going to the Macintosh and Rob is going to set up the projector, and I'm going to use Intermedia to give a view of hypertext and where it is in terms of the context of Vannevar Bush.

Touring Vannevar Bush

Bush's Vision

The first thing we have to look at is Bush's vision. His vision, as we all know, is the Memex. These are the actual pictures from the September 1945 issue of *Life* magazine, in which an artist rendered Bush's ideas with Bush's approval.

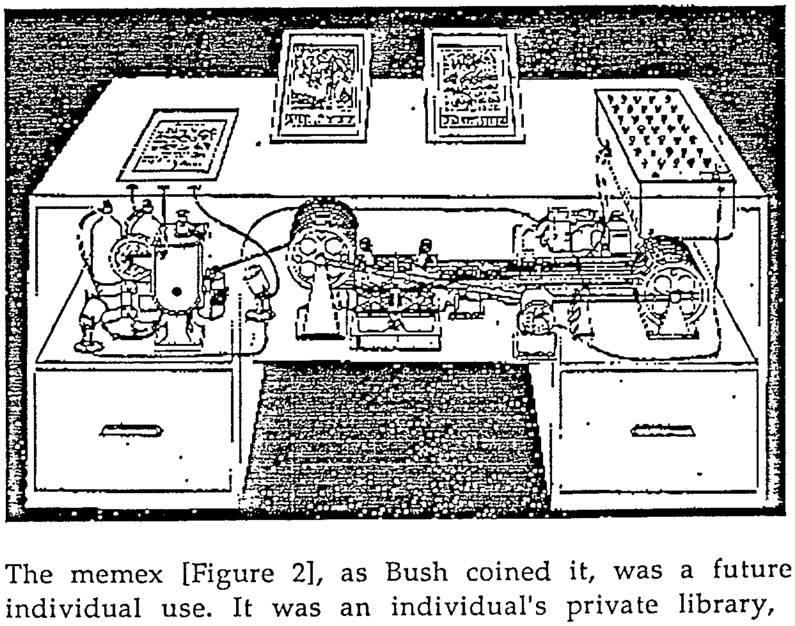

Figure 2

The memex [Figure 2], as Bush coined it, was a future device for individual use. It was an individual's private library, private file system, and as he called it, "an enlarged intimate supplement to his memory." And that was key — intimate supplement to memory. It was something personal, something with which you were truly engaged. As we move forward, these are very difficult goals to strive for, having something in which you can store all your records, store all your communications, and yet be so speedy and flexible. It's a big vision to live up to.

Technology

Pictures and Photos

As we look at some of the technology that Bush envisioned, we see that he thought that pictures and photos would be of great importance, that the "camera hound of the future would wear on his forehead a lump a little larger than a walnut." It would take pictures that would be projected, enlarged, and stored. In Bush's vision, one would take a lot of pictures for the record.

Where are we today? This is a picture of Bush's camera [Figure 3].

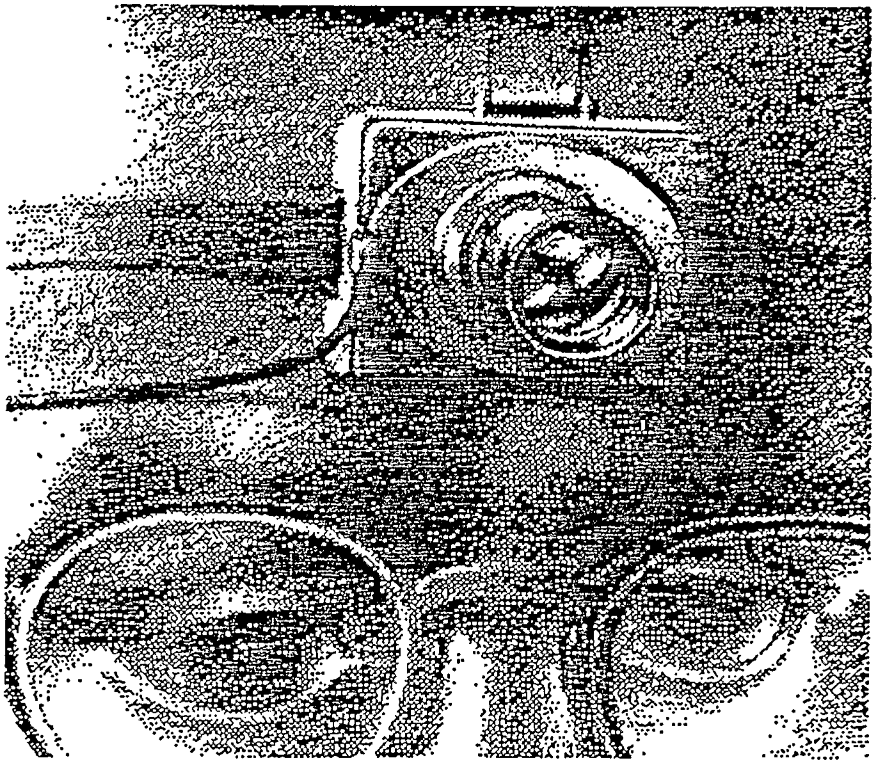

Figure 3

The closest we have to this right now in the hypermedia field is Ben Shneiderman taking pictures at conferences. But there are some technologies that are starting to look very, very promising. Instead of using film, which no longer is a fast enough technology for the magnetic/digital era, the Sony Mavica still video camera and the Canon still video camera allow you to take and capture photographs on a two-inch magnetic floppy disk. You can take those pictures and eventually display them on a computer screen. Right now it's cumbersome. It's not Bush's vision of taking a picture, plugging it into the memex and having it stored. Today you have to put the image up on a TV set, use a frame-grabber, and download the image to your computer. So it's a pain in the neck, but the basic technology is there to make photography much more integral, to be able to take pictures and have them go right into your record.

Similarly, the Sony 8mm Video Handycams are as tiny as a Japanese passport, with zoom, autofocus — the whole bit — and you can get four hours of video on a $5 videotape. So we have the technology for recording, but we still need the technology for storing and playing that back via computer. We're only halfway there, but we've made a lot of progress.

Voice

The next thing that Bush talked about was voice. Bush said that the author of the future would cease writing by hand and would instead talk directly into the record. But where are we right now? This was his picture of a Vocoder [Figure 4].

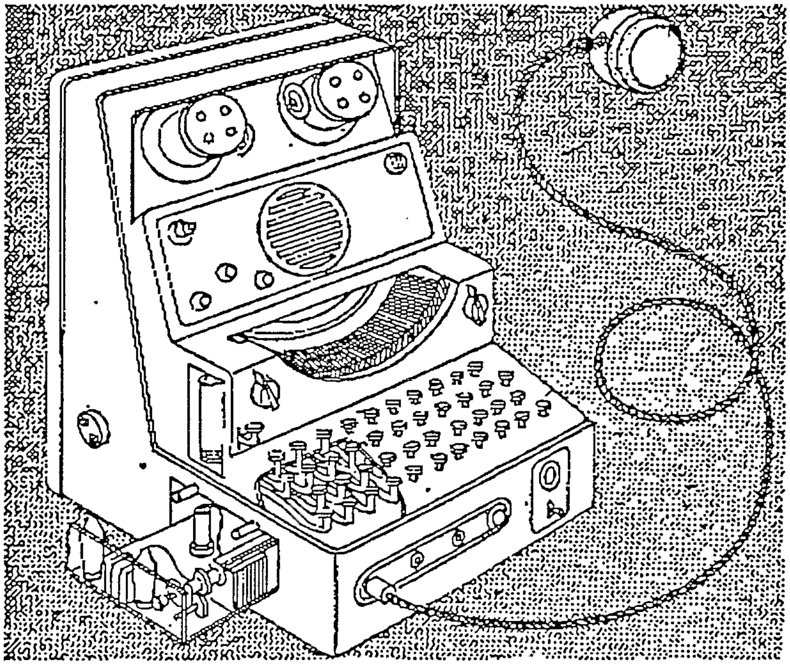

Figure 4

We actually don't have anything that looks like this right now, fortunately, but we are making some strides on voice recording. There's the Xerox PARC Etherphone experiments. Polle Zellweger and others will speak about how they've incorporated hypermedia paths into a system using voice recording that is quite nice. That's available right now on a research basis. There are now commercial products like Farallon's MacRecorder for the Macintosh and a new product that will do voice annotation. There's LipService, a voice electronic mail system on the NeXT machine, and there are the ubiquitous digital telephone voice mail systems that we all use, in which our voices get digitally recorded when we try to call somebody. (It's a little-known fact that nobody really exists in California, there are only digital telephone systems taking messages).

But we haven't yet solved the automatic transcription problem. There are many research projects that have realized basic voice understanding, but this isn't an area that has of yet been terribly fruitful. Physicians and lawyers, for instance, make great use of dictation, but until they can have a machine that converts dictation automatically into text, they're going to think this computer and hypertext technology is all a second-order effect. So if hypermedia is going to be provocative and we want to have a permanent record, you have to be able to speak the record, rather than type the record.

Wireless Networks

Bush also talked about wireless networks. He said "One can picture a future investigator in his lab. His hands are free and he's not anchored....He moves about and observes....Times are recorded automatically....If he goes into the field he may be connected by a radio to his recorder." How close are we to that vision?

There are currently a variety of portable computers you can get that have cellular phone modems in them. Without having to be near a telephone, you can log in at 2400 baud. There is also an intriguing technology coming from a company called Agilis — a DOS and UNIX portable with a packet radio Ethernet connection. By using the airwaves, without even the need for an FCC permit, you can now link at 10 megabits per second back to a server. This starts to become exciting. So we're potentially within a couple of years of having whole wireless networks where you are not tethered to your machine.

Handwriting

Handwriting recognition was another very appealing technology that Bush thought about. He said that the individual using the memex "can add marginal notes and comments....can take advantage of dry photography...and can [write handwriting] using a stylus scheme." We've replaced dry photography with scanning, which we'll talk about in a second, but where is handwriting recognition right now? Essentially, we're again within a couple of years of having some breakthroughs. Grid has a new portable computer with a stylus mechanism. You can actually write on the portable's and it will translate into text. IBM's T.J. Watson Research Center has shown versions of a stylus mechanism where you can actually fill in a Lotus 1-2-3 spreadsheet with a stylus, and it will automatically convert it to numbers. And there are other startups lurking in this area of handwriting recognition. If we can incorporate this very promising capability into our hypermedia environments, we will start to get the information environment that people envision.

Scanning

Scanning — where are we with mass data input? Bush wrote that "On the top of the memex is a transparent platen. On this are placed longhand notes, photographs, memoranda, all sorts of things. When one is in place, the depression of a lever causes it to be photographed onto the next blank space..." That sounds like a great vision, but where are we? Scanners exist, but by-and-large they're not personal yet. They're still shared, except for little hand scanners which are fairly cumbersome to use. The software for scanners is still cumbersome beyond belief. Bush talked about putting a piece of paper on a platen, pressing a lever, and it was done, like a copying machine; it went somewhere in the memex automatically. The way you use most scanners is that you open up an application, you preview the scan, you change the threshold, you scan it three times, you finally get it, you have to name it as a file, you forget the name, you can't find the scan anymore. Scanning is something you can't do with abandon. You have to do it very carefully because it takes a long time and takes a lot of storage. So we're not there yet, but we're close.

Similarly with OCR. There are many personal OCR programs which are pretty good, except they still take too long, two or three minutes per page, there's still too much intervention, and the accuracy isn't good enough. What you want to be able to do, if you're going to have this information environment, is take all the paper that you get in the mail, throw it in a hopper like your copy machine hopper, press a button, have it automatically scanned for you, timestamped, named and put in a folder. Later on you can do some information retrieval on it to see if there is anything of interest to you or browse through it very very quickly. You can't do that today. If you want to OCR something, you're very explicit about wanting to get one particular article. You'd really like to be less explicit, have have the system just read in all of your mail, first class or junk, and later only present to you the items that pass a particular filter. So we have some work here.

Bush also said "if the user inserted 5000 pages of material a day it would take him hundreds of years to fill the repository, so he can be profligate and enter material freely." Where are we in terms of that vision? I took a little liberty to do some experiments. If you look at bitmap scanning today, it's about 1 minute per scan; you can only do about 480 per day in an 8 hour day if you don't eat or anything like that. At 3 minutes per page for an OCR scan, you can do about 160 OCR scans per 8 hour day. So we're off of Bush's projections by somewhere between 10 and 50.

Storage

Now lets look at storage. If we do 5000 pages at 2000 bytes per page for machine-readable text, we're talking about 10 megabytes a day or 3.65 gigabytes per year. We can do this with optical disk or magnetic disk storage, but it's not all that inexpensive, though it's close. When we start getting into digitized graphics — if we're talking about low-res monochrome, 75 dots per inch, which is what a Macintosh screen is, for example — an 8 1/2 x 11 inch page/picture takes 64 kilobytes per picture.

You now need about 320 megabytes a day or 100 gigabytes a year, which is a lot of storage. And when you talk about hi-res color — 150 dots an inch with 8-bits per pixel, or the Macintosh 75 dot per inch resolution, but with 32-bits per pixel for true color — that's about 2 megabytes per page, 10 gigabytes a day, or 3.65 terabytes a year. If you're using 20 500-megabyte optical disk cartridges, which will give you 10 gigabytes, you have about a $5000 a day optical disk habit. There are some compression schemes that can get these numbers down somewhat, but as Frank said, I grew up technologically with Andy [van Dam], and he never ever was satisfied with the current technology, even when you got order of magnitude improvements over the past, and I learned the same impatience. Even today's optical disks are not as big or inexpensive as we might want them to be. So Bush was being somewhat optimistic, I think, in terms of where would be. Though we have made great strides in digital storage, there's still further to go.

In terms of storage, Bush said "The *Encylopædia Brittanica* could be reduced to the volume of matchbox. A library of a million volumes could be compressed into one end of a desk." I did some calculations, and I'm happy to report that the matchbox right here — and I believe these are the same matchboxes that Bush had in 1945 — is 27.24 cc and the CD-ROM right here is 13 cc. Encyclopedias like the *Brittanica* text, the *Compton's Encyclopedia*, *Grolier's*, and some other encyclopedias have been put on the CD-ROM medium, so we have reached Bush's desired storage density and solved one of his problems.

Access Bush also said that "A record, if it is to be useful to science, must be continuously extended, it must be stored, and above all, it must be consulted." One of the problems we have here is that even if we have the storage density — the CD-ROM matchbox containing the *Encyclopedia Brittanica* — CD-ROM is not necessarily the medium that is most appropriate. First of all, you typically can use only one reference work at a time because you have only one CD-ROM drive, and swapping CD-ROMs becomes a pain in the neck. The retail price of the CD-ROMs is still largely prohibitive. The price is at least $200 per CD-ROM for volumes that often cost much less in print (and even though the CD-ROM mediumitself costs under $5 to produce, even in limited quantities). Accessing the same information over slow national networks at large amounts of money per minute is not the right way of handling this storage problem, either. Just because we have the density of CD-ROM or optical disk doesn't mean we've solved the access problem.

Use

Selection Besides technology, Bush talked about use, and he was very interested in selection and association. He said, in a famous quote, "Selection [of that day]...is a stone adze in the hands of a cabinet maker." It was so hard to find something in the library, to find information — we weren't giving people the appropriate tools.

Where are we now? Today, manual selection of things is the paradigm that we use in desktop computing — pointing at something and selecting it as our focus. That is a very labor-intensive, manual effort, trying to find the focus by actually browsing through entire works. Search by content, finding information by its attributes or its semantic content, is still haphazard. And we still have the same filename problem that we've had for 20 or 30 years. Unless you are a very disciplined or compulsive user, it's very difficult to think of a name for something when you're first creating. Later on when you want to find it, you can't recall it because you didn't have the proper name for it in the first place. We are still locked into that mechanism today.

Association

In terms of selection and association, Bush had this vision of "two items to be joined, projected onto adjacent viewing positions. At the bottom of each are a number of blank code spaces, and a pointer is set to indicate one of these on each item. The user taps a single key, and the items are permanently joined."

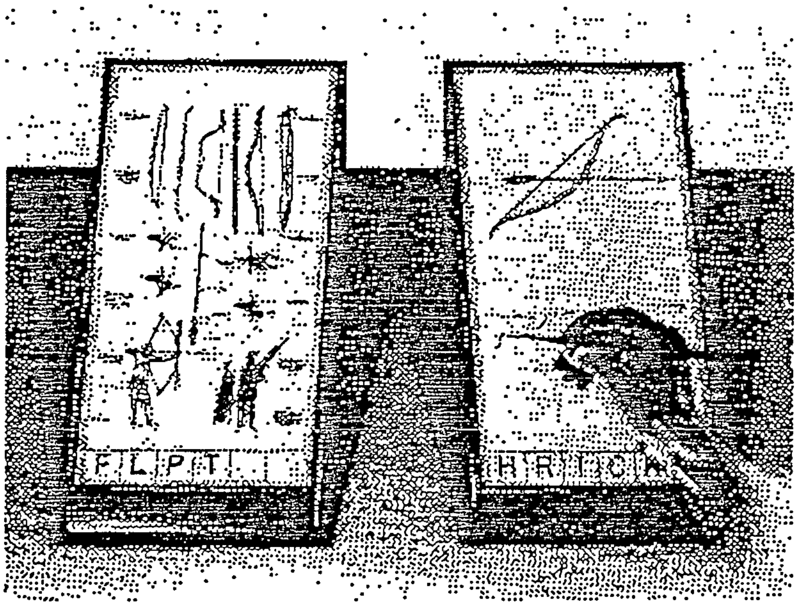

Figure 4

That sounds about as close to hypertext as we've been — this was his vision of the system circa 1945. Interestingly, even to this day there are an inordinate number of hypertext systems out there whose designers still haven't realized that when you want to link two things it is often essential to see them both at the same time. So multiple windows and multiple viewing surfaces are quite important, and it is unfortunate that that isn't seen as fundamental to all hypertext systems.

Furthermore, is creating links that easy? Is it so seamless in our systems? Is it done in the context of daily work? Bush was talking about being able to create links at will. Right now, with most of our systems you have to get into your hypermedia application, import the text from your text processor, import images from your graphics editor, and wade through a lot of dialog boxes. By the time you're done making links you've forgotten your original context. So we have a problem with the way hypertext systems are implemented today: they are not and cannot be used easily in people's daily work. They are used as special-purpose applications rather than as the primary tools with which people operate.

Automatic links

Bush also talked about automatic links. He said that "we can expect it [memex] to do clever things for us in the handling of the mass of data we insert it into it. We particularly expect it to learn from its own experience and to refine its own trails." There has been some work in that, but there are not very many hypertext systems that people can actually use to make automatic links. There's been some work on similarity measures at the University of Strathclyde. Jim Coombs at IRIS has done some work on using text patterns to link together various articles (which you can see in the demos). There's work on Bayesian inferencing by Mark Frisse at Washington U. and by Croft at U. Mass. and work on clustering by Crouch, et al. at the University of Minnesota. The latter three are in a session tomorrow. It's worthwhile to look into this area and see how we might put automatic linking into our systems and have them learn for us.

Social Issues

Contents/Materials

Bush talked about some of the social and cultural implications of the memex, of hypermedia. He said "most of the memex contents are purchased on microfilm ready for insertion. Books of all sorts, pictures, current periodicals, newspapers, are thus obtained and dropped into place." Again, where are we right now? I think we're not very far on that. By-and-large we have specialized, hard-wired content that you can get which comes along with specialized, hard-wired applications. You can get *Grollier's Encyclopedia,* or *MedLine,* or Computer Library, but they all have their own idiosyncratic interface. They're all very functional in and of themselves, but they're all very separate, and it's impossible to just buy data. You have to buy data and some program to read that data. So every time you buy another piece of data that's interesting to you, you have to learn a whole other interface and a whole other retrieval mechanism. Will we ever get to the point where we're not selling software, but we're actually selling just the content? That's a question we can ask ourselves.

Funding

Finally, Bush talked about funding. In his 1967 paper, *Memex Revisited,* he said, "The libraries still operate by horse-and-buggy methods, for there is no profit in libraries. Government spends billions on space since it has glamor and hence public appeal. There is no glamor about libraries, and the public do not understand that the welfare of the children depends far more upon effective libraries than it does on the collecting of a bucket of talcum powder from the moon." He didn't think that the memex would be done soon, but that eventually it would be created. So where are we to this day? Perhaps it's not the space program that's the scapegoat in 1989, but there are other programs that are taking over the funding that we should be using for building up our intellectual infrastructure, and it is still a real problem.

Why is this? Is hypertext so difficult to get funding for? There is no money and there is no glitz in creating an information network, but we need to have significant government leadership and scientific strategic leadership to make this a primary effort in our country, and internationally as well.

So What's Wrong?

This picture is not so bleak. A lot of the technology is just around the corner for putting this all together. So what's wrong with hypermedia? What do we have to do?

Does everyone recognize this picture of a rotary telephone? You may not, but this was part of the desktop infrastructure for 30 years, and it did not change very much. It stayed virtually the same for years and years and years. Only recently, did you get telephones like this feature-phone. This is actually my telephone, and I'll show you how to work it later on if you want. But it took a long time to get these new handsets that have much more functionality. Why was that? One reason was the whole notion of monopoly and companies not wanting to innovate. But I think another reason was that instead of going and working on the handsets, the "end-user applications," the phone companies spent much of the 60s, and 70s, and early 80s building the infrastructure, building the connectivity that allows you to make a call now virtually anywhere in the world, so that you can have long distance services that are virtually error-free and ubiquitous. Building the infrastructure is incredibly important.

The real question we have to ask ourselves is how do we build the infrastructure for hypertext? Where is this information environment going to come from? What is it going to look like so that we're not simply building fancy handsets but we're building handsets that connect to other handsets and give us this vast information environment that Bush was talking about?

The Desktop of Tomorrow

My contention is that if hypermedia is to catch on and secure its place in the computing repertoire, we have to integrate hypermedia features into the standard desktop environment that we are all used to today: the infrastructure has to be integrated into the standard computing environments and standard networks of today and tomorrow. There's an installed base of 30 million IBM PCs and compatibles and 3 or 4 million Macintoshes. It is fairly egocentric to have anyone in the field of hypermedia say "we're just going to create a new environment and it's going to be a wholesale replacement for all the existing environments that people are using." Instead we have to work somewhat "within the system" to create an exciting environment. But I think existing personal computer systems are set up so that we can create stimulating information environments. The next third of this talk looks at how all of this might work, how we might integrate hypermedia technologies into the desktop of tomorrow.

Requirements

What do people want to do, in terms of what Bush was talking about and in terms of hypermedia? They want to create content — graphics, text, and so forth. They want to reference things. They want to associate content. They want to have references not only to static media but to dynamic media. They want to have tools that tell them where they are in a vast information space. They want to do some filtering and querying and searching on that information. They want information processed automatically. They want to work in groups. They'd like to have some semantics to their information so it has some intelligibility. And they'd like to have some standard information services. I want to go through each of these topics and talk about how they might be addressed in the desktop of tomorrow.

Content

One of the things that we have to wrestle with is that the content that people want to deal with is diverse. These are the different types of information that I was able to fit on one slide:

Rich Text
Structured Graphics
Bitmap Graphics
Timelines
Tables
Video
Animation
Spreadsheets
Calendars
Cartographics

3-D/Rendered Graphics
Voice
Audio
Music
Statistical info
Modelling info
CAD/CAM
External databases
Reference materials
Symbolic math

and there are many more. The information that people want to link together ranges from CAD/CAM drawings to animations, from symbolic math to voice, from bitmap graphics to timelines, and to other things I have not even put down here. One of the questions we must ask is "can any hypermedia system can provide all of these types of content?" Can Intermedia provide all of those kinds of applications? Can Notecards? Can KMS? Can any? No. Why, not? Can any application developers, can any third-party developers, provide all the applications that are out there? If you look at the big players, Microsoft and Claris and Lotus, they can't provide all the applications. So it stands to reason that if they can't provide all the applications, neither will they be able to provide all the applications and provide a hypermedia environment, too. The only way we are going to get all the content editors that people want and all the hypermedia functionality they want is to have third-party developers create applications that handle appropriate content while hypermedia functionality is built into the desktop system just like cut, copy, and paste, so that all application developers can participate. Then you'd be able to link together documents from all the applications, all the content that you normally deal with.

Anchors

So how is this done? First, we have to add the concept of *anchors* to the desktop environment. People are very familiar with selection. Everybody seven years old and up knows how to select things on a Mac or on a PC with Microsoft Windows. So we have to extend the notion of selection to an *anchor* — a persistent or "sticky" selection. People already know how to create a selection, so if you say "hey, if you do this special thing, the system will remember your selection next time," as if you had run over the item with a yellow highlighting pen in your text. So that's the first thing we have to do. Any entity now, which they already know how to select, can be an anchor, a selection that's persistent, that's stored over time. The system can remember an insertion point or a word or a paragraph in text, or an object or a group of objects in graphics, or a cell or a row or a range of cells in a spreadsheet. If we have anchor functionality as a fundamental part of our system, then we have the basis for hypermedia. Anchors are essentially the next level of desktop integration; they're the next stop after selection and cut, copy, and paste. We want to be able to have anchors that have names, that have attributes so that we can do queries on them, and that can be the source or destination of links. So you're linking to very fine-grained things, not the very coarse card or document, but to whatever you've actually pointed out. Even if you don't want to have links to them, you can store these anchors as bookmarks, as important areas that you want to highlight permanently in your document.

Navigational links

The next thing we want to do, now that we have anchors, is to introduce the notion of *navigational linkage* into the desktop. Previously, we've had cut, copy, and paste. Everybody knows how to do that now. What we want to do next is have navigational linkage, where you can have a persistent tie between two anchors. If we have a user interface like cut, copy, and paste —where instead of saying "cut" you say "start link," instead of saying "paste," you select something and say "complete link," and a link is created between the two — it is very easy to teach people, because they already know how to do cut, copy, and paste. In this desktop architecture, we should make sure that links are bidirectional so you can both follow forward references and also find out who references you. Since any anchor may be a very significant anchor, like the beginning of the

Gettysburg Address, for instance, you want to make sure you can have more than one link emanating from that anchor, so it can be linked to many things. You want to have a very rich structure down at the desktop level, not a very basic one, so that users can model as many things as they like.

Warm links

From navigational linking we want to go to *warm linking*. We want not only to be able to follow a link from source A to destination B, but also to exchange data over that link. We want to be able to push data that's in a selection in document A into document B. Or vice versa — if we're in document A and there's something significant at the other end of the link, we want to be able to pull that data in, and have it replace our current selection. This allows us easily to create updates of information that everyone links to. If you have a central paragraph or list of telephone numbers that often changes, you can pull the new information in. The important part is that we can use the same mechanism that we use for navigational linkages for these warm linkages.

Hot links

Similarly, we want to use the same architecture to support *hot linkage*. Hot linkage is the automatic synchronization of anchors at two ends of a link, such that when you edit the information in one anchor, the altered information automatically is broadcast to all the other anchors that are linked to it, and is typically editable everywhere. That's currently being done partially in some individual applications such as Lotus's Jazz and Modern Jazz (which actually never was released) for the Macintosh. In Apple's System 7.0 there's a publish and subscribe mechanism, for Microsoft Windows on the PC there's DDE (Dynamic Data Exchange) and Hewlett-Packard's New Wave. So we're beginning to have mechanisms that allow us to have this hot linking, this hot copy and paste. One very difficult problem is editing any type of element in any other document. If you want to be able to paste the bottom line of a spreadsheet into a word processing document and be able to edit the bottom line of the spreadsheet in the word processing document, suddenly you have to have editing capabilities for spreadsheets in your word processing application, or you have to have some object-oriented protocols and building blocks such that when a piece of spreadsheet data gets put into a word processing document, all the editing facilities and all the capabilities come along with it. Carnegie-Mellon's Andrew system and HP's New Wave begin to have these capabilities, but are not in widespread commercial use. Yet as object-oriented programming and those type of architectures become more prevalent in the systems that we all use, it hopefully should become common commercially within the next two or three years.

Active Anchors

After you have this hot linking, again using the same basic mechanism as navigational linking and warm linking, you have to start looking at dynamic links, or links to *active anchors*. Today's applications, by and large, are passive applications. You essentially bring something up, you edit it, it immediately gets updated, and the control is returned to the user. Newer applications like animation, video, music, and voice are temporal in nature. They exist in time; they're not static. Creating links to them and creating links from them is a very complex problem. Anchors now have to represent time spans and one has to start creating standard mechanisms and standard policies so that when a user follows into an active anchor, the appropriate thing happens. For example, when the user follows into a video clip, do you want the clip automatically to start running? Do you want it to position at the beginning of the clip and wait for the user to issue the play command? Do you want the author to be able to specify that option? These are all issues that need to be worked out, but we have to have some consistency and some policy that works the same for all temporal applications. There is some work going on in this area. We've done some work, Randy Trigg has done some in the VideoNoter project, and Tim Oren has been looking at some of those issues in the HyperCard context.

The dynamics in anchors is not only limited to the temporal axis. Especially for model-driven animation and rendering, the anchor has to reference not a temporal span, but a set of viewspecs that will allow the destination view to be computed at follow time with the same parameters as were used when the link was made.

Wayfinding

Wayfinding capability is the next thing that we have to put in the desktop environment, and this is not simply for hypermedia. It's for all the desktop, because separate from hypermedia, people still have these same questions: What's my current state? What have I done right now? What have I done in the past 5 or 10 minutes? And what can I do next? Where can I go? What's available to me?

History

Those wayfinding questions can be answered by a variety of mechanisms. We need to add *history* and *paths/trails* to the desktop environment. For history, even separately from hypermedia, the system should remember all of the windows you have on the desktop and all of the ones that you've opened that session. If you look at the PC and the Macintosh, every application typically has its own "window" menu that tells you all the open windows for that application, but not for the other applications. You spend a lot of time switching between applications and then switching between window menus finding what is open and what's not. I'd like to have a standard mechanism: one window that gives you the history of everything that's been opened, everything that's been closed, everything that's been followed into, everything that's been activated and so forth. This history window should be included as part of the desktop environment, and should include hypermedia information, but not just hypermedia information. That's very, very important.

Paths/Trails

Paths and trails are very important (as we'll hear in the next session from Polle Zellweger and Cathy Marshall). There have been some nice breakthroughs in how to do those paths. Paths are essentially histories that have been captured, edited, shortened, and made into concrete desktop objects that can be played back. Again, we're trying to look at keeping within the desktop metaphor. The important thing about paths is that we'd like users not to have to program these paths but to be able to create these paths by actually doing the traversals and editing out events that they don't necessarily want to keep a record of.

Maps The next type of wayfinding that we need is *mapping*. There has been some work done on maps, and the general conclusions are that global maps of large hypermedia systems don't give people particularly useful information. If you have a binary tree approach that spreads out visually, you don't see enough of the information, because the information isn't topological and you can't create a global map that has any continuity; the information is all over the place. So rather than spending a lot of time trying to create global maps, which become unwieldy with massive amounts of information, we should concentrate on creating local maps. These give you the details of your local environs and how you can branch one level or two levels out from your local environs. It's easy to generate these local maps, and they give people good cues as to where they can break out of their current environment. If you are in a city, you typically do not take out a world atlas to find out where the open restaurants are, you take out the local city map (although, as you all have discovered, it's a trick question here — there are no open restaurants in Pittsburgh on Sunday night). Again these maps should be an intrinsic system structure. They should be generated as you open the desktop, and as you open new documents, they should automatically be updated. People get familiar with them; they're there all the time. This is not something that you have to batch compile, or make on the fly, or draw yourself; it's done for you by the system.

Filtering We want to make sure that with hypermedia we do not fall into the trap of thinking that it can solve all problems. We have to have a nice, tight coupling between hypermedia linkage/browsing technology and information retrieval technology. We want to be able to apply filters to our hypermedia on system attributes such as author, creation date, and modification time and on user-supplied attributes and values, so that you can start issuing some complex queries. We need to have *collection filtering* where we can run a query over an entire hypermedia web yielding a list of all the hits that fulfill the criteria, and *exposure filtering* where we hide or display the icons that indicate anchors and links based upon the filtering criteria. So it's the Engelbart *viewspecs* notion, where we can have various things exposed based upon criteria that we issue. We have to have these notions of filtering in the desktop so that all of our information can be filtered, hypermedia information being just one of the types of information that we care about.

Queries

We need to do something about queries. We need to have easier-to-use interfaces for creating queries. The state of queries for end users is a total mess. Not because there aren't good interfaces. In fact, there are a hundred good interfaces, and that's the mess. When you get any database program or any full-text retrieval program or any large database that's out there on a Macintosh or a PC, it comes with its own idiosyncratic interface and its own strategy for doing boolean expressions and for specifying keywords. It's time there was a standard, system-provided query sheet that individuals could fill out, save as a concrete object on the desktop, and open up to have a query issued automatically, accessing a diverse set of databases with the same user interface. You no longer would need separate interfaces for every type of database you had. One of the problems with such a strategy, however, is determining how much power one needs in the query interface. One ends up in large arguments: Do we have to give the users all the power of SQL? Do they have to have nested queries — "give me all the documents with this keyword that have anchors in them with that keyword that have links in them with this other keyword"? Or do we need to have much simpler queries that allow users to get better feedback very quickly? These are the very knotty issues we have to tackle. We have to start standardizing on some of these things so users can actually start using our systems.

Content Searching

We have to start looking at *content searching* as well as the keyword-type filtering that I talked about. We must see that full-text searching is compatible with hypertext, and together they are much synergistic and more useful than they are separately. Again, you want to have full-text indexing as a system function. So for any application, as long as it has text some in it, whether it be a graphics document, or a spreadsheet, or a text document, the text is indexed. And it's indexed behind your back. There is no full-text database into which you have to force your documents. Behind your back the system is reading documents as you change them and computing an index for you, so that you can find all the documents that have particular words in them. Having that system anticipation — where it indexes things before you explicitly ask — is liberating. You get some positive fallout from having full-text retrieval that helps in the next feature that I'll talk about, *virtual linkages.*

Virtual Links

One of the things that Frank Halasz mentioned in his Issues paper at Hypertext '87 is this notion of virtual linkages. Often you don't want to explicitly, manually create the anchors at both ends of the links. Often you'd like to anchor something at the source but say that the resultant documents and anchors at the other end of the link are going to be the result of a query. Well, if you have full-text indexing you can all of a sudden say here's a link which I anchor to some source, and when I follow it, a query is issued saying, for instance, "find me all the documents that have the word 'lipid' in them." The system then does a full-text search on all the documents, system-wide, to find those that meet the criteria.

Well, if you don't have full-text indexing, if you want to have a virtual link like that, the system has to painstakingly search through or grep through (for all you Unix hackers) the content of each document in the system. If you have a full-text index, you can start to have virtual linkages that occur instantaneously, because the system simply looks at the index, not the actual content of all documents. That becomes very, very exciting. The synergy between full-text indexing and virtual linking is quite significant. Again, we want to have these virtual linkages in the desktop environment as part of the linking technology that all applications participate in, with full-text in there as base functionality as well.

Automatic links

Automatic linkage is something else that we'd like to have in the desktop environment, though this might be a little further out than the previous features. You'd like to have the system. while you gone for the evening, bring new information into the system (from mail or newswire, for instance), forge links in the background based upon patterns, or Bayesian inferencing, or clustering, and notify you in the morning. You'd like to have different options for this. The system might just do the whole process automatically. Or perhaps it should allow the user to review each link as it is created. Or perhaps it should make all the links but replay them through a history mechanism and let you get rid of the ones that you don't like.

Group Work

Finally in terms of some additional hypermedia features, we'd like to have some group work technologies. We'd like annotations, like notes — the Post-It™-like notes that have become prevalent in a lot of hypermedia systems. We're seeing a little of this already. Applications like MacDraw, spreadsheets like Excel, all have little note facilities. Rather than having a hundred different application-specific note facilities that work differently, let's have note facilities for leaving little pop-up notes as something that's standard on the desktop, so you can create a note on any application with the same user interface. Importantly, and this doesn't occur with all note implementations today, you want the notes to keep track of their anchors in the document as you edit the document. You want to make sure that your annotations stay in synch with all of your content. A lot of the annotation programs you can buy in the market turn your document into a bitmap and you can attach notes to the document. But you're not attaching notes to a live document, you're creating notes to a dead document. While behind your back somebody else is editing the real document, your annotations no longer reflect the state of the real document. We have to use hypermedia technology that keeps our anchors correlated with our notes as we edit the document and create annotations.

Similarly, we want to have annotations that allow us to draw or write notes on top of documents much like we would use an acetate sheet, and again, we want those acetate marks — those proofreader's symbols or whatever — to work on editable documents. As I edit a document, the marks move about the acetate to keep in synch with the content to which they are anchored.

As I mentioned earlier, we'd like to integrate voice as a desktop medium. NeXT has done that in their mail system. We want voice to be easily used in the hypermedia context as well. Essentially voice is an application, and you want to link to and from voice just like any standard application, using the standard, system–provided linking protocols. Another area to explore in hypermedia and group work is using hypermedia linkages in the desktop environment to handle the threads in mail and conferencing systems.

Semantic Markup

Beyond pure hypermedia issues, there are other things we want in the desktop to support this information environment. Semantic markup is one of them. Right now, our WYSIWYG interfaces are by-and-large procedural. There is a lot of manual labor in changing rulers, moving objects around, changing patterns, and so forth. We'd like to start moving away from that, still using a direct manipulation interface, but beginning to have more semantics of document style. Just as Scribe had in batch systems, you'd like to have standard, in all direct manipulation systems, document style sheets so that documents of a particular type can be created from templates that are already filled out. You can start having object-oriented document classes, where you have, say, a Personnel document, and you can create subclasses of that for UnionEmployee or NonUnionEmployee. Now where does hypertext come in? We'd like to have hypermedia templates as classes. This is similar to the notion of composites that Frank talked about in his talk two years ago. You want to have links set up automatically between different documents, so now you can create a class of document that already is prelinked to another class of document, and when you instantiate a class you get an entire web of empty template documents but with all the links pre-made. Once we can start doing that, you won't have to manually forge every link that you create, but rather have the system understand some of the inherent linkages between documents.

Services

Finally, we want to have some *services* in the desktop of tomorrow. Besides *linking* services, other services that are important are *reference* and *linguistic* services. You want to have the reference tools that you're used to, like dictionaries and glossaries, integrated into the system in a standard way, with a standard dictionary protocol so that all dictionaries can be accessed through the same type of interface. You'd like to be able to have multiple dictionaries and set up a dictionary path so that when you select a word and issue the "look up" command, the system follows a path you have set and tries to find the word in all the dictionaries you have put in your dictionary path, your virtual bookshelf. You want to have a way for people to make their own glossaries. Often work groups have their own terminology, their own acronyms, and people should be able to create glossaries in a group, and have them work just like the real *American Heritage Dictionary* or the real encyclopedia. You want the glossaries to use the same protocols and the same mechanisms, so your personal dictionary is part of the dictionary path you normally use. Of course, we want thesauri to use that same mechanism.

We'd like to have spelling correctors and grammar correctors in the desktop. Today all the spelling correctors that exist are application-specific, and you have to learn 27 different spelling correctors to use all of the different applications you normally run. All of these utilities and reference tools should be part of the base system, with third-parties supplying the actual data and specialized-extensions.

We'd like to have morphology services. Right now, when you do information retrieval and type in a word like "filter," it will typically get "filter," and sometimes "filtering." We want to start using morphology, linguistic analysis where you can get "refilter," and "unfilter," and all of the morphological derivatives of that word, which will give much better hits from information retrieval queries. It might be nice to have standard pattern-search mechanisms. Again, if you look at today's applications, every single application that's around has a different interface for doing pattern searches, but by-and-large each is doing the same thing. We'd like that interface to be a standard in the desktop.

Challenges

Hypermedia fits very well in the desktop as the next level of integration, but it needs a lot of support if it is going to be exciting and important. So as we enter the final part of the talk, we want to look at the challenges.

Challenge 1: Let's start putting the theory into practice. Let's build and analyze the systems that we create, especially at appropriate scales, and test them out on users, not just ourselves.

One of the challenges we have is that our field is very compelling to ponder. Many fields have too little vision, and in some sense, our field has too much. We can hear and see and taste and smell the future so well that sometimes we forget to build it. Just as the perfume that comes in the airline magazines isn't the real thing, we have to make sure that when we're talking about having large, multi-user hypertexts, that we are building those things, and not just doing proofs of induction where n = 1 and that's it.

Challenge 2: Input and ouput mechanisms must be central to my daily work for hypertext to take hold.

The next challenge is the technology challenge of input and output. As we saw, scanning, OCR, handwriting recognition and annotation, and voice input and recognition are all getting there. We have to start working with the developers of these technologies to get them to recognize that these devices need to be intrinsic and integral to our system, so that they work seamlessly. You can't create systems that have all of the peripherals working as if they just met the CPU today. They have to be intimate, so the scanner, the OCR, the handwriting recognition, work well with the system. It's a matter of having hardware developers and peripheral developers work closely with the hardware platform developers and system software developers. So peripheral is a misnomer. Input/output technologies must not be peripheral to the system, but rather central to an integrated environment.

Challenge 3: Make the screen display exceed paper. Until individuals choose a computer over paper for reading, hypermedia will not catch on.

Richard Saul Wurman is one of my favorite graphical information designers. A page from his *London Access* guide is dense with information represented by text, graphics, icons, symbols, maps, and color. A page from the new *U.S. Atlas* he just did provides new clarity for road atlases by using crisp graphic images coupled with sound organization. One of the interesting things about the atlas is that it was done entirely on the Macintosh using desktop publishing. But it was done in a way such that you could never present this density of information on the screen. You can print it out to get color separations and so forth, but the technology just isn't there to completely display this sophisticated information on the screen. One of the challenges is to make the screen display meet or exceed that of paper output.

Right now desktop presentation is still pretty lousy. Computer Graphics folks typically focus on and get their jollies over how long it takes to make one screen, rather than how short at time it takes to generate a screen of information, and we have to try to change that attitude around. As Nick Negroponte said, rendering tea kettles is not high on the list of what most people want to do in this new information environment.

The low density of screens, even though they're much higher than they used to be, still is not good enough to present information like the Wurman slides I just showed. We need to have flatter screens and much higher resolution, at least 150 dots an inch, better color, and so forth. We need to have portable screens that have better resolution. We're not there yet. Hypermedia is only going to catch on if the screen exceeds paper.

Challenge 4: We need a removable storage medium of the 90s. We need cheap, random access, exchangable, ubiquitous storage — the gigabyte floppy.

We must not think that we have all the storage problems solved. CD-ROM and CDI technology is a weak technology for interactive computing. It's a great technology for transmitting information because CD duplication simply involves stamping out injection-molded plastic, which is much cheaper to do and much faster to do than recording magnetic or optical media. But we don't want to be condemning all data to be read-only and non-malleable. We want individuals to be able to annotate and manipulate the information. We need to start making sure that we have a ubiquitous medium that is as dense as CD-ROM but is also very interactive. We need to have removable medium of the 90s just like the 3 1/2" disk of the 80s. But it has to be a 1 Gigabyte floppy disk that is cheap with which everyone can transmit information, store information, and link information.

Challenge 5: We need a common anchor model. Just as all applications now handle selections, all applications must handle anchors.

As the next challenge, we have to devise a common anchor model. We have to start seeing anchors as fundamental parts of the system. We have to come up with standard ways in which to train users to create anchors, and standard ways to indicate the existence of anchors, just as we came up with ways to indicate non-persistent selections.

Challenge 6: We need to work on multi-user issues. As researchers, we need to better formulate the requirements and solutions for sharing.

We have to look at multi-user issues. Hypertext has been going through the emperor's new clothes syndrome, where we talk about it as a multi-user technology and a wide-area technology but typically the systems are single-user systems that aren't networked and don't solve many of the multi-user problems. We have to move from workgroup to local-area network to wide-area network, and concentrate on the problem of shared hypertexts. We need to look at a problem that has not adequately been addressed: that of shared keyword indices and authority lists, and how to handle conflicts. Most indexing done today is done by professionals, using large authority lists, who typically divide up a large corpus and work on separate parts. If you look at the results, there are many inconsistencies between indexers. If we now have shared hypertexts, and we want to have multiple individuals create a single index, we need to provide a model for all individuals to see the shared keyword taxonomy, to notice conflicts, to manage synonyms, and to create their own private taxonomies when the group taxonomy doesn't meet their needs.

Similarly, it's still unclear how we should deal with document transfer and exchange. If I create an item here, and create a link from it and send you a diskette, do you have copy of the information that was linked to it or just a reference? As for the source document, is there any way to know what the genealogy of the information is, where that document was first created, so that if I receive a modified version of my original, it can be automatically merged back in? These multi-user problems have not yet even been solved at the small group level, much less at the local area or wide area level, so there is much challenge ahead.

Challenge 7: We need transferability and sharing through a nationwide and international file system, with transparent internetwork addressing, file addressing, and anchor addressing.

We also have to address the multi-user issues from the wide-area perspective. How do we begin to manage document addressing and uniqueness? Do we use a model like the proposed Nelson/Xanadu back-end byte stream protocol? There is a new project called the Collaboratory that Bill Wulf at the National Science Foundation is getting underway. It proposes to create a national collaboration network, a "center without walls," hooked together through a transparent nationwide and international file system. Regardless of the base technology that is used, we have to begin to deal with the creation of a national and international infrastructure allowing true addressing to the anchor level.

Challenge 8: Apply object-oriented techniques to hypertext.

How do we make hypertext be something other than unstructured programming, a melange of gotos with no coherence? Structured programming helped slow down the proliferation of spaghetti code by introducing a stricter notion of subroutine calls and a standard "methodology" of entering and exiting routines. Object-oriented programming went a step further by allowing users to model objects that pointed to other objects, and coupled the notion of data structure with data *behavior*. Classes serve as templates from which objects of like structure and behavior are manufactured, and from which objects with similar structure and behavior can be refined/subclassed. We need to look at object-oriented techniques for hypermedia. Documents are the end-user objects, and links are the pointers connecting objects. We need to look at document classes and webs from which linked structures can be spawned and refined. And we have to look at knowledge lattices and taxonomies as ways to organize such classes in an intelligible fashion.

Challenge 9: We need funding for long-term research in the hypermedia and information arenas.

There are some social challenges. How do we get funding for this area? It's a massive research project of national and international import, but government research support for this area is tiny. There's a bill before Congress sponsored by Al Gore, for a $1.9 billion high-speed national network largely for supercomputing. How do we make sure that this is not only for supercomputing, but forms the basis for a national network that is speedy enough to support a hypermedia information environment?

How do we convince those with the funding of the need for longer-term research? Right now you can get funding for short-term work in hypermedia, but that's largely from companies saying "we'll give you some money to port Intermedia to the X Window system on our Bazinga Workstation in the next three months" or "Can you create an electronic version of this particular manual by next quarter?" That's not going to further the field, that's just going to help some companies gain visibility or produce a product with a very short shelf life.

Challenge 10: We must encourage trailblazers and allow the scaffolding and rhetoric of hypertext to evolve by rewarding such work professionally.

Bush talked about the profession of trailblazers, who would create the links between records and would be the modern-day editors in the hypermedia world. You'd be able to see not only the content that an expert creates, but the scaffolding from which it was created.

The problem is that there are few experts today who are actually making those linkages, because there aren't hypertext systems they can use in their daily work. The rhetoric of hypertext still isn't all that well understood, although there are beginning to be some good efforts. There's a new book coming out by Bob Horn from Information Mapping Inc. that has some very nice discussion about the rhetoric of hypertext and there is a talk later in this conference by Fitch RichardsonSmith from Pittsburgh about graphic design for hypermedia.

The other problem is that junior faculty members do not get tenure for creating hypertext linkages, and so the freshest scholars, and often the people with the most creativity, do not get rewards for creating hypertexts, and in fact are strongly discouraged from doing so. How do we allow the scaffolding and the rhetoric to flourish, to be rewarded, and to be a professional activity?

Challenge 11: Re-examination of copyright laws to achieve parity between rights of author/publisher and needs of community.

The copyright issue...Will copyrights replace ambulances as the things that lawyers chase? The copyright issues have just not been addressed. I don't have any particular details about how the laws need to be changed, but there needs to be a concerted effort to re-examine these issues by those people who have both the necessary technical expertise and the legal expertise. They need to examine the copyright law and make sure that the owners of information have their rights, but that the community of people who need to use that information without profiting directly from it can have their rights, too.

Challenge 12: Let's understand the policies we are trying to capture before standardizing on a mechanism to capture it.

There are the challenges of hypermedia standards. Standardization is a good thing at appropriate times, but we shouldn't standardize on something if we don't know what it is. We shouldn't standardize on hypermedia until we understand hypermedia *policy.* A lot of people standardize on *mechanism,* and then soon realize that it was the policy — how you use things — not the mechanism for storing or representing them, that they needed to standardize on. X Windows is a great piece of technology, but they standardized on the wrong problem. Everyone has agreed on X Windows as the mechanism for display windows and graphics, but there is total discontent about which user interface package, which policy, should go on top of that. They haven't standardized on anything, because they're all fighting about what package is going to sit on top of X Windows. We shouldn't do that in hypermedia.

SGML is an important standard, but we should make sure that we don't stretch it too far. We should make sure that it really works. Often standards get created and they don't get used. A lot of applications can write out SGML subsets, but very few of them can read full SGML back in. We need to make sure that standards are fully operational, and easy enough that they can get widely implemented.

And then there are other standards organizations that are downright dangerous. They just like making standards. If there was half a need for a "ransom note" standard, they would be gathering a committee right now.

So we should standardize, but we should first make sure that we're standardizing the policy, and not just the mechanisms, of hypertext.

Challenge 13: Publishers should make available, for next 2 years, machine readable copies of their holdings for non-commercial research in hypermedia by qualified institutions.

Publishers own materials and copyrights on those materials, but there is little research in building hypermedia versions of the materials. Publishers are primarily interested in creating real products immediately. If we don't have material for hypermedia researchers, if we don't have hundreds of big works which researchers can put into a large hypermedia corpus, we're never going to understand the issues of generality and scale, and we're never going to be able to push our systems forward. It is to the publishers' advantage to let researchers have access to a wide-range of materials for research purposes, so that researchers can push not only the technical attributes of hypermedia systems, but so can foster breakthroughs in the process of creating and linking materials as well. So I challenge all the publishers here, and all the publishers who are not here, to make available over the next two years at no cost to qualified hypermedia researchers large amounts of content, so we can actually experiment with what we need to sustain long term hypermedia development from the system side and the materials development side.

Challenge 14: Linking protocols from all major vendors within the next 2 years. By Hypertext '91, all major platform vendors (IBM, Microsoft, Apple, NeXT, *Sun, Digital, HP, etc.) should supply application-independent protocols for hypermedia and the information environment.*

The final challenge is the "missing link." This is a summary of what I've been trying to propose throughout the talk.

Today's monolithic hypermedia systems are compelling, they're reasonably easy to build and design — as we can see from the more than 20 demos scheduled this evening — but in some sense, they're doomed to failure because they're just one more application off to the side.

The current level of integration is the desktop metaphor and cut/copy/paste. It is prevalent and has caught on because it's easy to learn, it was functionality that everyone wanted, just as hypermedia functionality is desired today, and it was integrated deep in the system with a protocol provided to application developers in system-level toolboxes that allowed them to add cut/copy/paste to their applications at very low cost.

If hypermedia is to survive and blossom, the major vendors need to provide system-level hypermedia support, where you have standard linking protocols that all developers can participate in, providing support for anchors, navigational, warm, and hot linking, wayfinding support for maps, paths, and history, support for keyword and content search and filtering, automatic links, and virtual links.

We no longer want separate hypermedia applications. We want a linking protocol deep in the system. We want the applications that everyone uses today to be the nodes of the hypermedia system. We want to move from the era of the clipboard to the era of the linkboard, where documents from the regular applications that we use every day — Lotus 1-2-3, Excel, Microsoft Word, Autocad, MacDraw, WordPerfect, etc. — are the entities that we can link to and from.

So the challenge, by Hypertext '91, is for all major platform vendors, the IBMs, the Microsofts, the Apples, the Suns, the NeXTs, the Digitals, the HPs, whomever, to supply application-independent protocols for hypermedia in their systems, so that we can have the current system architectures as the basis for the information environment of the future.

Conclusions

That bring us to the question "Hypermedia — Does It Reduce Cholesterol, Too?"

There are a couple of answers. One is that sources tell me that, yes in fact, HyperCard Version 2.0 will have a cholesterol reduction feature. I'm not sure how reliable those sources are.

Another way to answer that question is to say that just as oat bran is important for reducing cholesterol, so is hypermedia important for reducing information clogging and information glut. But just as diet of nothing other than oat bran Twinkies is not the ultimate key to health, neither is a diet of nothing other than hypermedia the key to our information well-being. What we need is an information environment, in which hypermedia is an important and fundamental element, but is just one of a balanced diet of elements that, together, bring us towards the vision that Vannevar Bush set out for us in 1945.

Acknowledgements

Julie Launhardt, Marty Michel, Andy van Dam, and Nicole Yankelovich kindly helped me turn this audio transcript into something that more closely resembled the English language. Karen Smith Catlin, Tim Catlin, Ron Weissman, Paul Kahn, Marty Michel, Andy van Dam, Bern Haan, Nicole Yankelovich provided significant critique, and all my colleagues at IRIS helped formulate the ideas presented in this paper. I am grateful to Wayne Jacques, who provided me with a tape of the almost unrecorded last third of my talk.

References

Bush, Vannevar, "As We May Think." *Atlantic Monthly*, Vol. 176, No. 1 (July 1945), pp. 101-108.

Bush, Vannevar, "As We May Think: A Top U.S. Scientist Forsees A Possible Future World In Which Man-Made Machines Will Start To Think." *Life*, Vol. 19, No. 11 (September 10, 1945), pp. 112-124. (Condensed from the Atlantic Monthly, July 1945 with illustrations by Alfred D. Crimi.)

Bush, Vannevar, "Memex Revisited," in *Science Is Not Enough*, Vannevar Bush. New York: William Morrow, 1967, pp 75-101.