

PCEXpo 2001, New York City  
Keynote Address  
Thursday, June 28, 2001  
10:30 AM EST

## **Living in a Distributed World: Mobile Computing and the PC**

By David Gelernter, Ph.D

The file system is dead -- that permanent bureaucracy that grows inside all our computers like crab grass. The desktop is dead too. In other words, our core information management tools are ALL deceased. Lots of people know these systems are dead, but a fair number of them don't want to tell anyone, because they figure that the file system and desktop work perfectly for everyone else...

Maybe THEY can't find what they're looking for, sometimes; maybe after inventing their first 20 million or so ridiculous file names they've got galloping file-fatigue, which unfortunately is not covered by most medical plans -- folder-fatigue is a closely related disease; but they assume that everyone else is doing just fine.

The real problem is that, in today's computer world, there's no such thing as an "everything structure" -- a single data structure in which your whole electronic life is stored; every electronic document or application you own, everything you've ever created or received.

Instead of "everything structures," we use the Virtual Tupperware system. Real Tupperware works fine, but unfortunately there are significant differences, it turns out, between electronic data and leftover meatballs.

Our electronic documents are scattered by the thousands in all sorts of little containers all over the place. Some are in some folder, some are on the desktop; some are in mail files, some are images in their own special Tupperware boxes, some are calendar notes, some are bookmarks in the browser; some are on drive C and some are on D, some are on the laptop, or the palmtop, or the computer at work, or the Macintosh you reluctantly unplugged five years ago but couldn't quite bring yourself to throw out. That's a lot of virtual Tupperware, and we're getting more all the time.

The more information and the more computers in our lives, the more of a nuisance this system becomes. It gets steadily harder to locate things. Eventually we arrive at the three-card Monte approach to information retrieval, or the shell-game technique: the challenge is to figure out what shell your file is hiding under. Too often, you lose. Even more important, today's conventional approach means that your cyberlife is scattered as if you'd emptied a file cabinet out of an airplane; there's no everything structure; there's no one place where you can see the whole story - - all your information, your whole cyberlife. There's no single place where you can see immediately what's new, where you can search or browse everything at once; there's no single structure you can tune in from any net-connected computer anywhere to get access to everything.

There are no everything structures; therefore, there are no "subset of everything" structures, and there's no good, clean way to SHARE information either. There's no clean way you can hand someone access to "everything I've got about the Zeppelin project" or "all my dealings with Joe Feinstein," because there's no good, clean way you can get access to such a thing yourself.

\*\*\*\*\*

I'd like to discuss one view of today's core information management tools, and how they got that way; and I'll also discuss a new and different approach to information management.

We call our "different approach" a "narrative information system." The basic idea is simple. You go through life creating and receiving pieces of information, electronic documents. You can give each document a name and file it away as it arrives. OR you can let all your documents accumulate in an ongoing time-line or narrative – a kind of documentary story of your life.

The "electronic documents" are the same in either case -- all your files, memos, email messages, photos, scanned images, bookmarks and so forth; everything. But we treat them very differently depending on whether we see them as files in a file system or as episodes in a kind of narrative.

If they're files, it's natural to organize them by folders; if they're episodes of an ongoing story, it's natural to organize them not into folders but into themes or sub-stories -- not the Zeppelin project FOLDER but the Zeppelin project STORY, or the story of the Piffel Report, or the story of your dealings with Joe Feinstein.

It's natural for a file to belong to exactly one folder, but an episode can appear in many different stories.

It's natural to classify files one at a time as you get them, but you put stories together retrospectively when you need them.

It's natural for files to be stored in some particular place, but a story isn't a physical object; you can tune it in wherever you are.

A file system is a static structure. An ongoing life story changes and evolves; it has a past, present and future.

Your "documentary life story" is an Everything Structure. In the long run, some sort of Everything Structure is a prerequisite to successful mobile computing. When you're on the move you usually have less time to play where's-the-file with your computer, and less screen space to play it in. When you're glancing at your computer instead of staring at it lovingly for hours, it's especially important to be able to tell quickly what's new and what's coming up -- you need to see the recent past, the present and the future all at once, right away. And you need to be able to reach back as far as necessary to find anything you want. Ultimately, mobile computers won't succeed by guessing the exact 7 functions you need when you're on the move, or exactly the right 12 categories of data you've got to have...

What DO you need when you're on the move? Everything, fast. You need access to everything; you need to be able to find anything you want immediately.

Our requirements are simple: on every computing device you own -- every desktop, palmtop, cellphone and solid-state coffeepot -- the information environment should be EXACTLY THE SAME, subject only to the physical limitations of the device and its display. If you know how to find a file or email on your computer, you should know how to find a stored phone number on your cellphone. And if the phone number is AVAILABLE on your computer, it should be available on your cellphone. To look at it from the other direction -- finding a stored number on a cellphone has obviously got to be a fast, simple transaction. You don't have the time to waste mucking around. Finding a file or email on your desktop computer should be just as fast and simple -- Because when you get right down to it, you NEVER have the time to waste mucking around.

Once you can drive a car, you can drive any car. Once you can drive a computer, you ought to be able to drive any computer -- including all the cellphones, palmtops and other special-purpose computers in our lives.

\*\*\*\*\*

Some people say that file systems will never be replaced by narrative information management because file systems are too old and engrained, and narrative systems are too new. In fact we don't expect and don't want narrative systems to REPLACE anything. We layer them on top of today's software environments, without changing or disturbing the pre-existing software in any way; nowadays, that's the way software environments evolve.

But the philosophical point is worth looking at. However old and engrained file systems may be, they're a lot less old and engrained than life stories. Few ideas in our lives are more basic, more natural and more "organic" than narratives and timelines and stories.

Traditional file systems made sense during the opening phases of the computer age. They're complicated; they make users do the filing, organizing and retrieving. But computers didn't used to be terribly powerful, and it used to be a good idea to make users do as much of the work as possible. It takes a powerful machine to achieve simplicity. Nowadays computers are powerful enough for the simplicity of narrative information systems.

First I'll discuss today's conventional approach; then I'll discuss this new "narrative" approach. Narrative information management is the basis of a commercial system called Scopeware built by a company called Mirror Worlds Technologies. It would be crazy to predict that Scopeware will emerge as THE winner in the changing world of information management, but I will predict this: Scopeware's properties are the properties that a winning system will need, and the direction Scopeware points is the right direction.

\*\*\*\*\*

My topic is not only where this software stands today but where it's going, and how we think it will get there. Distributed computing in general, peer-to-peer in particular play a big part.

On the one hand, we'd like a future in which everyone's cyberlife is available on any net-connected platform -- your own machines AND the computer at the supermarket, the airport, the ATM lobby. But we'd also like a future in which we USE the enormous power of the desktop PC, not flush it down the drain by turning our desktop machines into mere window-frames we reach THROUGH to get to the real action on the Web. So we can't merely build an Everything Structure and park it at some web site. The desktop computer must be central, not peripheral. A future that's rich in powerful PCs that are used merely for Web browsing is a future where everyone owns a Ferrari and every speed limit is 15mph. It doesn't compute.

How do we meet these requirements? We're long-time believers in distributed computing. When we built our first implementation of Linda and the tuple-space model of distributed computing in the mid 1980s, we used a technique we called "uniform distribution instead of servers"; today it's usually called peer-to-peer. We continue to believe that distributed computing in general, peer-to-peer in particular, are the basis of the cyber-future.

\*\*\*\*\*

Today's core information management is based on three related components: named files; hierarchical folders or directories; and the desktop. These are your basic tools for dealing with your own information world. All three are obsolete.

It would be a miracle if they WEREN'T obsolete. File systems first appeared in essentially today's form in Unix, during the 1970s; Unix was loosely based, in turn, on a cleaned-up version of Multics, a huge operating system of the late 1960s. Linux is of course a direct descendent of Unix.

Microsoft's Windows desktop first succeeded in version 3 of 1990; it was based on the Apple Macintosh operating system, which appeared in 1984. Apple's system was based in turn on the pathbreaking Alto system invented at Xerox during the 1970s -- but Apple made its own significant changes to the Xerox design.

In short, nothing much has happened in core information management since 1984.

By the early 1980s, all the major pieces were in place.

We had the file system with named files and folders.

We had the desktop with icons and windows and menus and the mouse.

We had all that in 1984, 17 years ago, when many of today's best programmers were still in nursery school.

When -- statistics suggest -- 14% of today's technology world CEOs, 59% of leading technology reporters and 81% of our nation's hardest-driving venture fund executives were still wetting their pants.

(I got those statistics by making them up, but you get the idea.)

We rely for our core information management on 1984 software. We do NOT rely on 1984 hardware. How many 1984 PCs do we use today? If you could FIND a 1984 PC outside a museum, it's unlikely you could convince your pet dog to use it. No modern house pet could get by on 64 kilobytes of main memory. You'd have a cruelty-to-animals case on your hands.

Desktop computers from 1984 do NOT satisfy our computing needs today. But we rely every day on information-management SOFTWARE from 1984.

And relying on this obsolete software is even stranger than relying on 1984 hardware would be. The information world has changed radically since '84. In 1984, email was a small-scale phenomenon. Computing power, memory and disk storage were scarce and expensive -- many PCs had no hard drives at all. There were no or almost no electronic photos or faxes or MP3s or digital scanned images or animations or videos to deal with. And of course, there was no Web. This was the environment that our 1984 information management software was designed for.

It was brilliant at the time. It helped make desktop computing succeed.

But how can we blame it for not working well any more?

It was designed to operate in a radically different, radically more restricted, radically smaller-scale technology environment than the one we have today.

\*\*\*\*\*

Still: what's wrong with these tools? They're old but that doesn't automatically qualify them as obsolete. They've EARNED the right to be called obsolete.

In today's standard file systems, every file needs a name and no two files in the same folder can have the same name. These rules are a pure unnecessary nuisance.

All sorts of documents don't want or need names.

Why should a shopping list have a name?

Or a letter, or page of notes, sketches or calculations, or a scanned-in image?

Why should a snapshot or rough draft or reminder note have a name?

Only formal documents need names -- papers and essays, books, poems, reports; but INFORMAL electronic documents are growing more and more dominant as computing becomes more and more universal.

Your information management software should let you name documents only if you want to; only if the names make sense.

And it should let many documents share the SAME name.

If you've written a series of drafts of the Zeppelin Report, it's natural to call each one "Zeppelin Report draft," or something like that. Requiring that each draft have its own name is ridiculous.

Our rigid every-document-needs-a-name rule has many bad consequences. We make up names that are nuisances instead of helps. When your file system is cluttered with documents called "letter to Joe," "new letter to Joe," "new new new new new letter to Joe" and so on, something's wrong.

And these rigid naming requirement put significant restrictions on the power of our software and computers. For example: lots of important business gets done by email, more all the time. When I search for something, I want to search email alongside all my other files. But instead, I need to dive INTO the mail file and muck through my email separately. Why? Because an email message has no name. Therefore it can't stand in the file system on its own two feet. It has to be dumped inside some OTHER file that DOES have a name. Email messages are permanent second-class citizens, merely because they have no names.

When we let unnecessary, obsolete restrictions dictate the way we do basic business on a computer, something is wrong.

\*\*\*\*\*

In today's standard file systems, every file must belong to some folder or directory. No file can belong to many folders. These rules are another unnecessary nuisance.

How many people do you know who keep every paper document they own in a folder? (And what kind of people are they?)

Some documents belong in folders and many don't. Again it's especially the all-important INFORMAL documents that often don't.

And it's clearly ridiculous that we should allow a document to be stored in only one folder; paper documents are physical objects that can only be in one place at one time, but no such restriction applies to electronic documents. You should be able to put an electronic document in as many different folders as you like simultaneously.

And why, for that matter, should you put ANY documents in folders? Why can't the folders reach out and take them? Paper folders can't do that trick, but why should our software be limited by the capabilities of paper?

Working together, named files and hierarchical folders have turned modern desktop computers into electronic versions of 1940 Steelcase file cabinets. WE do the filing, we do the organizing, we do the retrieving. Why do we put up with it? How many of us wanted to grow up to be file clerks?

As we work to file things properly and figure out where we've put them, the computer just sits there pretending to be a piece of furniture. This is a silly way to use powerful electronics.

Today's standard desktop is a bad match to modern requirements too. Just the fact that the desktop display is STATIC, stays the same unless you take the trouble to change it, makes clear that it's no longer an adequate tool.

What IS the desktop? A picture of your electronic information world. But your electronic information world is always changing, and the desktop is always staying the same. New mail and messages arrive all the time; you create new documents all the time. When you look at your information management system, you should see the up-to-date story as of right now.

There's a deep problem behind all these core information systems. The hierarchical file system and the windows-based desktop were invented separately, and have never gotten along very well. You ought to be able to look directly AT your information system, with no desktop or any other separate, special-purpose interface standing in the way.

\*\*\*\*\*

Some people say -- so what? Today's tools may be inadequate, but we're stuck with them. They're too engrained. This is wisdom and experience talking.

Only a wild-eyed ideologue or -- even worse -- an academic would fail to grasp that people are just too heavily invested in these systems to change, ever.

Of course, Wisdom and Experience made exactly this same argument to explain, in the early '70s, why the IBM mainframe and DEC minicomputer would rule the world forever.

People used the exact same compelling argument to explain why the PC and the desktop workstation were just toys and would never catch on.

They used the same argument again to explain why DOS was here to stay; why windows and mouses were just toys and would never catch on.

But once Microsoft Windows arrived, they were positive: the technology world had now stabilized forever. We were dug in and that was that.

Browsers and the Web were just toys and would never catch on.

And by the way, Java was just a toy and would never catch on; C, C++ and so on were too dug in.

Ecommerce would never catch on; no one would ever send a credit card number over the internet.

And the list goes on.

In retrospect, the only thing that IS permanently engrained, that we ARE permanently committed to forever, is this argument.

You can say this for it, it's been a consistent performer. Wrong nearly 100% of the time.

Why do we keep making this argument when it keeps being wrong? Good question. I don't know.

Maybe narrative information systems will triumph and maybe not, but we know this for sure: the file system and the desktop as we know them today are history.

They're not good enough, they don't meet today's requirements and they are on the way out.

That doesn't mean that the companies that build them are finished, not at all; but the technologies themselves are through. They were great in their day, but the party is over.

No one and nothing, no power on earth, can keep a technology onstage when the party is over.

\*\*\*\*\*

What do we suggest instead?

Instead of today's systems, we suggest narrative information management -- based on a software structure we call a "stream": a flowing, growing, time-ordered river of electronic documents.

I'll say a little about what the stream is, what it looks like, how it works, how a public stream or company stream or community stream relates to your own private stream, and what it offers instead of the standard static desktop and the hierarchical file system.

\*\*\*\*\*

The stream is exactly your "documentary life-story," your narrative history, your cyberlife; your Everything Structure.

It captures all your documents of all kinds.

Whenever you create or receive a new document, it goes at the head of your growing stream.

So your stream is a time-ordered parade of documents, beginning with your oldest -- in theory, with your electronic birth certificate -- and continuing through "right now" and on into the future. A "document" is any file, email message, bookmark, memo, draft, scanned image, calendar note, photo, shopping list, voicemail message, rolodex, business card; anything.

When you tune in your stream, using any net-connected computer, ordinarily you see a receding parade of documents onscreen. The closest documents are nearest in time. When a new document arrives -- when you get a new email message, for example -- the message appears at the head of your stream -- at what we call the NOW line. When another email shows up, the new message takes its place in front, and the rest of the documents in the stream all take a step back -- so in this sense, the stream "flows." Ordinarily you stand on the NOW line looking from the present into the past. Farther-away documents are older. The farthest-away is the oldest.

When you create a new document -- say the first draft of a report -- you put an empty box at the head of your stream, and then you put whatever words you want inside (using any word processor you like).

When you scan in a document, we OCR the image and attach the text version; this new two-part document, image-plus-text, appears at the head of your stream and then flows back, with all the rest.

When you schedule a meeting for 2PM tomorrow, you put a note in the future of your stream. Your calendar note flows steadily towards NOW. When it reaches NOW, at 2PM tomorrow, it appears at the NOW line and then flows back into the past, along with everything else.

You can see the whole future by "turning around" and looking at it; you can see it in the form of a flowing stream, or as a conventional calendar. Your calendar is just the future of your stream.

\*\*\*\*\*

The "receding parade" is an efficient way to present information on a computer screen.

It uses foreshortening -- the perspective effect when you look at a road, say, head-on instead of from the side -- to pack maximum information into minimum space.

"Easy browsing" is an essential feature of the stream display; when you touch a document in the stream with the cursor, a summary with a thumbnail appears immediately in a small window next to the parade.

You control the stream with VCR-type controls -- move forward or back, go to the beginning, to "NOW," or to the end of time -- the stream's farthest recorded future; or you can time-travel to any date, past or future.

The stream-type display is important for mobile computing. For the time being, mobile computing centers on small screens: electronics may be getting much faster, but pockets aren't getting a lot bigger. The conventional desktop interface depends on breadth -- on letting information spread out. On mobile devices, there's no room to spread out. The depth-based stream display is a better match to small screens, and it's easy to tune in your stream using a palmtop or Web-enabled cellphone.

\*\*\*\*\*

Narrative information systems try for UNITY: they take an INTEGRATED approach to information.

ALL your information is on your stream, and you can tune in your stream from any net-connected device anywhere.

There's another kind of unity at the foundation of these systems: the system IS the interface.

There's no distinction between the shape of the information-storage structure and the shape of the interface -- no distinction between a folder hierarchy on one hand and a desktop landscape on the other.

The goal of a narrative-system interface is simply to show you your stream in whatever way is best for the device you're using and your immediate needs; in the simplest, most straightforward way possible.

In fact the stream is such a simple structure, you can easily picture it in your mind -- and that makes stream systems ideal for the no-screen computers that will become important: wearable computers, in-car computers -- computers will start being important to drivers when and only when they lose their screens -- obviously; phone-accessed computers, and other devices with audio instead of video interfaces.

It's easy to say "start from NOW and tell me the newest five elements on the stream," or "tell me the next element in the future," or "move forward one," "move back one," "search for Zeppelins and read off the result stream"...

\*\*\*\*\*

Ordinarily an organization maintains a stream for all its employees. At Mirror Worlds Technologies, for example, there's a "company stream" and we're all tuned in virtually all the time. We each have our own personal, private view of the company stream.

Your own personal stream -- your private view of the shared stream -- includes your email and all your other private documents, which are available only to you. Your view ALSO includes "public" documents, and documents belonging to any groups you're part of. So: everyone looks at the public stream through a private lens; you see your own personal material shuffled together with public information. The stream captures your own personal documentary life story, and also the company's life story. When you're tuned into the stream, you're tuned into the company. The stream captures the company planning its future, reflecting on its past and generally talking to itself.

\*\*\*\*\*

Your personal stream has the characteristics of a narrative information system as I described them earlier. When new documents arrive, you don't name them and file them away; you just sit back and watch them show up at the NOW line and then flow back into the past. The stream is less like a standard file system than like your own personal CNN.

Of course you don't have to get rid of your conventional desktop display when you get Scopeware -- you don't have to get rid of anything -- but the Scopeware stream offers of a very different kind of display than the conventional desktop.

The Scopeware Stream and the Desktop are each the front surface, so to speak, of a much larger information world. But in the case of the stream, the whole information world is stream-shaped - you can reach back as far as you want into the past or the future; the display always works the same way; and the desktop is a kind of still photograph that doesn't change unless you take the trouble to change it; the stream is a moving picture.

But you don't ONLY watch it; you also need to use it. If we're relieving you of the job of naming and classifying new documents as they arrive, we need to compensate by providing powerful tools to let you find what you want when you need it.

You find documents in the stream using a combination of three methods: SEARCH, BROWSE and TIME-ORDER. We find that the symbiosis of these three makes them worth more than the sum of parts.

When you SEARCH a Scopeware stream, you get a stream back -- a "substream" of the original. This is a narrative information system; your main stream is the complete, comprehensive story; when you do a search, you get a substream -- in other words, a focussed sub-story.

You can search on any word or phrase (every word in every document is indexed), on document types and meta-data, and on time-related data.

If you search on "New Haven Bottling," the resulting substream is a story or narrative or "documentary history" of all your dealings with The New Haven Bottling Company -- first contacts, subsequent internal documents or mail, reports and so on. Whenever you do this search, you get the all-inclusive, up-to-date story automatically. This "story" includes the future: if you're having a meeting to discuss New Haven Bottling next Tuesday, a conference call next Wednesday and a presentation next Friday, those calendar notes from the future of the stream are all part of the New Haven Bottling story -- they're all part of the substream returned by SEARCH.

\*\*\*\*\*

Fast browsing enormously increases the power of SEARCH-- makes it possible to sweep through a screenful of documents fast -- just touch each document-card with the cursor; you don't have to OPEN a document to get an idea of what's INSIDE.

Fast browsing makes searching faster and easier: you don't have to focus a search down to a single document, or a small handful; if search comes back with a couple of dozen documents, that's good enough. Using easy-browse, you can scan through them in a few seconds.

Time-order is always an underlying guide. Often you have a general idea of when the documents you're looking for showed up -- yesterday morning, last week, last July, whenever -- and you can use that information to look directly in the mainstream, or to home in on what you want after a search.

So a SUBSTREAM in Scopeware plays the role of a folder or directory in a conventional system. Instead of a "Zeppelin project" folder, you have a "Zeppelin project" narrative or story or, in software terms, substream.

The substream can do what a folder can, but it's more powerful.

A substream collects documents automatically; you have to put documents into a folder by hand, one by one.

A substream persists: it continues to trap new documents that belong to it.

(If you look at the Zeppelin substream tomorrow, you'll find that it's grown to include any newly-arrived Zeppelin documents.)

A substream tells a story, and it can include the future.

A substream is "non-exclusive" -- one document can be part of many different substreams.

A "folder," in short, imposes all the obsolete, irrelevant limitations of old-fashioned cardboard.

A "substream" on the other hand is an organizational tool designed for computers instead of clerks and file cabinets.

\*\*\*\*\*

A few short examples of the Scopeware stream in action, and a closing note on the future.

The stream has turned out to be a powerful tool for work-based, company collaboration. If I have something to say to the Development group, for example, I don't have to send out the message in a mass emailing, or put it on a bulletin board that people may or may not look at sometimes.

I put it on the stream, permissioned to the Development group; it shows up at the head of every developer's stream.

It stays on the stream, and anyone can go back and check it any time.

It can generate a thread of comments and discussion, all on the stream; this discussion becomes part of the company's ongoing life-story.

If I'm interested later in Zeppelins and Zeppelins were part of this discussion, the relevant comments will turn up along with everything else when I do a SEARCH.

By watching the stream, you're always up to date on what's happening.

But I can always put reference and archive material into the stream also; the reference documents are always there when you need them.

The company shares all its contact and address information automatically, simply by dropping "contact cards" into the stream. We can use the future as well as the present for collaboration: we can remind people of a meeting or deadline or company event by dropping a note in the future.

Stream-based collaboration has taken some unexpected forms. Our company stream is full of scanned-in menus from local New Haven restaurants: you can search on some pizza variety and get menus for all the local options. All sorts of discussions go on in the stream.

People in the New Haven office, people in New York, people at home, people on the road all read the stream, and it ties the company together.

By reading the stream, you read the company.

Another example:

Consider the "family PC." A major question the industry has mainly not bothered to answer: how does a family share a computer, or a bunch of computers? The stream-based answer is simple. There's a family stream, and everyone has a private view of it. You can tune in the family stream from any computer in the house. You see your own documents, email and so on, and there's shared material on the stream too: family appointments and plans are in the stream future. Family photos, for example, are dumped right into the stream; everyone can see them. The stream is a perfect photo album -- a browsable, NARRATIVE photo album that tells a story; photos are interleaved with everything else. You could store digital music and videos on the stream; that way, as your collection grows, you actually stand a chance of finding what you want.

Postings could be permissioned just to parents, or for that matter just to kids -- a great tool for fomenting conspiracies of all kinds.

But you can foment some family history too.

You could scan a young child's artwork or schoolwork into the stream; years later, when you actually wanted to see it, you'd be able to find it.

You could get your bills delivered to the stream, and always know where THEY were.

In short you'd have a family calendar, communication medium, multi-media archive and database AND a complete, unfolding family history -- along with each person's own private work space -- all as part of one simple, mobile-accessible system; a FAR simpler system, we'd argue, than today's desktop-file system combination, and far more powerful.

One last example.

There are dozens of reasons why many or most of today's web sites would work better if they were built as Scopeware streams, or with a stream right behind their front doors. But I'll mention just one specific issue. Many sites need constant updates, daily or more often.

News sites are a special case, fascinating from the Scopeware viewpoint, but I'm thinking now instead of companies communicating with their employees, schools with their staffs and students and parents, universities, summer programs of all kinds, labs and institutes and all sorts of offices, local restaurants, local theaters, all public places that need to tell their users or staff or paying customers what's going on; what's new.

These are, collectively, some of the hardest-working, most-consulted web sites in the world.

As I say, they need constant updating; but too often, updates mess them up and make it hard to find what you need, OR they're a nuisance to do and don't get done as often as they should be.

These sites sometimes go for days or weeks without updating, just because the update process is such a nuisance.

These sites ought to be Scopeware streams.

To update a stream takes about five seconds. Take your update; drop it at the head of the stream. That's it.

The update gets indexed and thumbnailed automatically.

Visitors see it immediately; it's in front.

Everything else in the stream takes a step back, but the rest of the stream is still visible and you can still find anything you want.

You can find things by search and browse, and the site can also include pre-programmed substreams on a menu -- like pre-tuned radio buttons, to take users directly to interesting topics.

You can throw your site map away.

This system is ideal for frequent updates; also, it lets you report in the future as well as the present -- report what's coming up, not in a separate site-category but as part of one stream; also, you have a complete archive -- anyone can look anything up at any time. The stream reaches back as far as you want it to.

We have a stream at our mirror worlds site, at "scopeware.com" or "mirrorworlds.com"; to find the stream, tune in the site and then click on the somewhat mysterious purple label "Search with Scopeware," at the top. The next thing you see is the stream.

If you browse the web with this stream structure in mind, and ask yourself how many other sites would work better if they were streams instead of whatever they are -- electronic cob webs, mazes full of secret passageways, advanced exercises in digital navigation -- the results might surprise you.

\*\*\*\*\*

Where are we today and where are we going? Our plans center on distributed computing, especially the peer-to-peer flavor.

To use a narrative information system today, you install a Scopeware stream on a server and tune it in from any net-connected computer, or from a palmtop device or Web-enabled cellphone. We're continuing to expand the range of wireless devices we support. Our first goal was to make the stream available on any device anywhere; so our first version -- today's version -- is server-based.

Our next goal is to bring the power of the desktop into the equation. Since 1984, core information management software hasn't changed; desktop hardware has grown more and more powerful. Today's computer industry faces a fundamental question: why should users continue to throw out their old computers and buy powerful new ones if we can't think of anything useful for the new ones to do? The old ones were idle most of the time anyway...

Why do we need 2001 computers if we're wedded to 1984 information management?

Narrative information systems, on the other hand, are NOT 1984 software. We admit it: they would never have worked on the 1984 Macintosh. But they WILL run, easily, on today's desktop machines; they give those powerful machines something useful to do in core information management.

Desktop power will continue to grow, and streams will continue to grow -- in the variety and sheer volume of the information they hold, and in the sophistication of your tools for dealing with them.

Before long, for example, our Scopeware streams will be virtual landscapes that you can fly over freely at any altitude to get the big picture... where you can time-travel into the stream's past or future at the flick of a joystick.

We're just completing our first desktop-based system. It uses the power of the desktop machine, and gives us close integration with the familiar desktop tools.

Our next goal is to reconstruct the shared communal stream as a VIRTUAL stream.

Today's desktop systems show you your personal stream -- all your own documents and communications.

But you also need to be able to post a public document to your desktop stream, and have it show up automatically in everyone else's desktop stream.

You need to be able to post a document to the Developer's group and have it appear on every developer's desktop stream.

We can do that today by combining the desktop for personal documents with the shared server for public and group documents.

But we can also build a virtual shared stream: when I add a public document to my desktop stream, the system automatically sends a copy to the other ten thousand members of the public -- or however many there are.

On each of those ten thousand other desktops, the document shows up and gets slipped automatically into the local stream. It APPEARS as if everyone is sharing access to a public stream, but the shared stream is virtual and not actual.

Of course we don't throw the server out; we want it for back-ups, archives, robustness. In fact there are all sorts of intermediate points between our pure distributed, peer-to-peer approach and the pure server version. We can rely on the server for public postings and on peer-to-peer for smaller groups; there are many possibilities.

The distributed version makes it especially easy to support quick, low-cost pick-up groups. If five people have some topic to discuss this afternoon, it's easy to create a group and a substream for five people and one afternoon.

[Then, the next step beyond our distributed desktop-based version: there's a group of machines you rely on regularly; your stream should be up-to-date on each one. All we need to do is replicate each stream posting to every machine in your "primary work group." That will require a lot of communication, a lot of computing, a lot of storage. And that's what cheap, powerful technology is FOR: to USE, not to hoard. In this new "primary work group" version, you can still tune in your stream from ANY machine anywhere, just by dealing with your stream remotely as we deal with all streams today, in client-server mode.]

We have lots more plans, too; but enough for now.

\*\*\*\*\*

In conclusion...

Over the last few decades technologists, especially hardware designers, have accomplished amazing things. Today's computing hardware is spectacular. It's so impressive and so good, it can jump through so many hoops and do so many wonderful tricks, we sometimes forget that the basics of information management don't change. We think and talk and listen; we read and write and look at pictures. That was the essence of information management in the 13th century and it still is today.

Computers can put us in touch with the world and, more important, with ourselves. On today's dense, complicated, fast-moving scene, our most important technology need is to be firmly connected to ourselves: to our own pasts, presents and futures. Computers can be our global switchboards and our open-ended, constantly-growing personal libraries. They can hold our unfolding life stories. But we need the right software. Our hardware is doing its job but our core information management software is not. It's time to get back to basics, and to get the basics right.