Literary MACHiNES

LITERARY

THE REPORT ON, AND OF, PROJECT XANADU
CONCERNING
WORD PROCESSING,
ELECTRONIC PUBLISHING,
HYPERTEXT, THINKERTOYS,
TOMORROW'S INTELLECTUAL REVOLUTION,
AND CERTAIN OTHER TOPICS
INCLUDING
KNOWLEDGE, EDUCATION AND FREEDOM.

MACHINES

by Ted Nelson

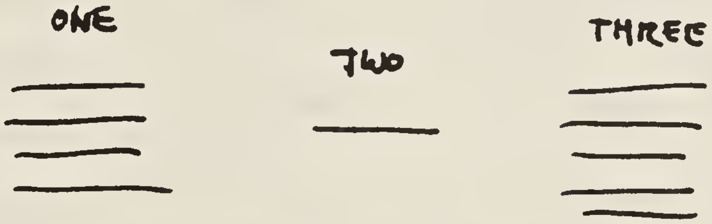
THIRD EDITION. This volume incorporates changes made by the Xanadu working group for the second edition, particularly the back-end specfications as described in the final chapter.

Anyone wishing to interface to the system should contact the Xanadu working group at 313/663-3637, as the details described in that chapter are not yet locked.

Additional copies of this book may be had by sending a valid check for fifteen dollars to Ted Nelson, Box 128, Swarthmore PA 19081. No purchase orders or deferred-payment plans can be accepted.

This book is a hypertext, or non-sequential piece of writing.

It is partly <u>about</u> hypertext, or non-sequential writing, and using a hypertext form will, I hope, help communicate some of the benefits of such writing.



PLAN OF THIS BOOK

There are several Chapters One, one Chapter Two, and several Chapters Three.

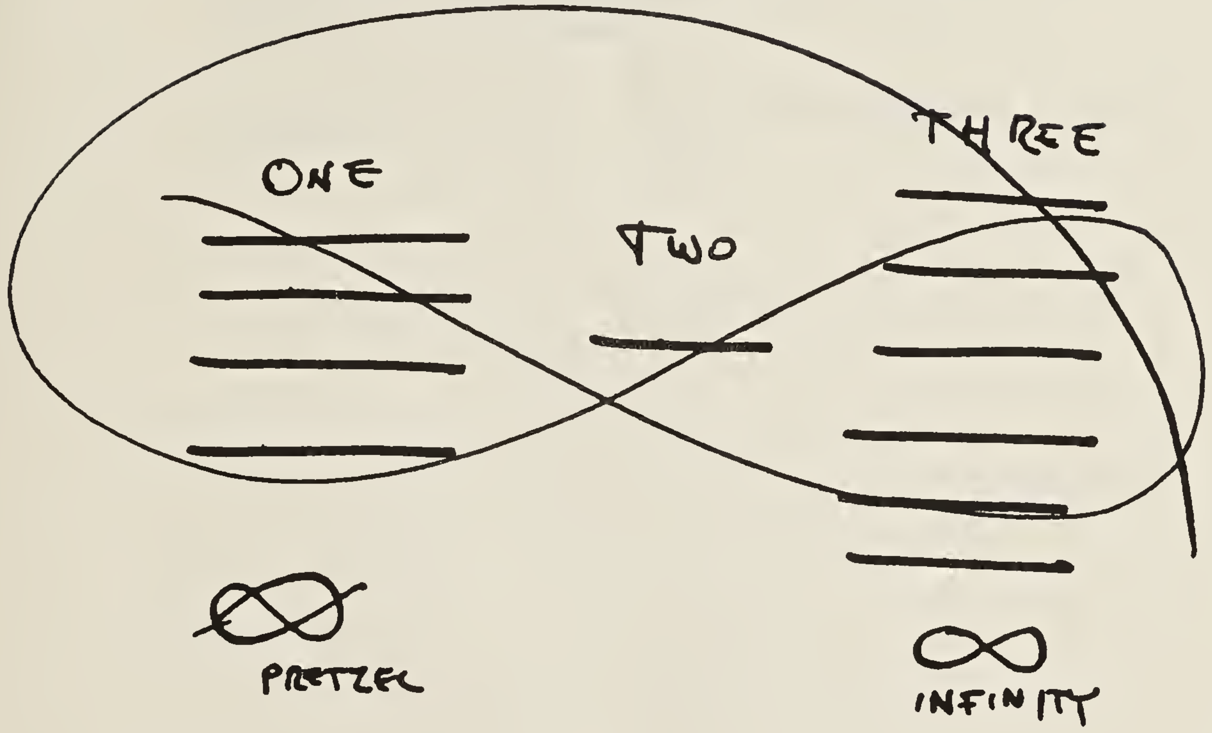
It is suggested that you read any of the introductory Chapters One first, and then Chapter Two; which is the heart of the book. You may or may not feel that you understand it fully.

It is suggested that you then read one of the closing chapters. This will help you see what the future of the system is supposed to be about.

At this point it is suggested that you read another of the introductory Chapters One, and look over Chapter Two again. You will almost certainly understand it better.

Continue in this vein, passing repeatedly through Chapter Two, until you understand this book.

Pretzel or infinity. It's up to you.



Whenever a work's structure is intentionally one of its own themes, another of its themes is art.

Annie Dillard

There are also several Chapters Four, which deal (insofar as possible) with certain technical aspects of the system.

No instruction for reading these chapters is provided.

DEDICATION

This book, and the system it foretells,
are dedicated to

Eric Blair (1903-1950)

better known by his pen-name

"GEORGE ORWELL";

an acute, sad and bitter
observer and prognosticator
who understood tyranny perhaps better than any tyrant,
who understood information control
long ahead of the rest of us;
and who left us cunning, elegant and timely warnings.

Somehow many take his name to stand for
all that he despised;
so that the word "Orwellian,"
meaning tyrannical, oppressive, mind-controlling,
and futuristically threatening,
is itself the perfect example
of the twisted Newspeak he foresaw.

**May his simple, honest, angry devotion
to truth and human freedom
live forever.**

Give me a lever long enough
and I will move the world.

Archimedes,
as generally
misquoted

Words without thoughts
never to Heaven go.

Hamlet

Extremism in the
defense of liberty
is no vice.

Barry Goldwater

Litera scripta manet.
(The written word remains.)

Horace

Toto, something tells me
we're not in Kansas anymore.

Goldwyn's Dorothy

CHAPTERS ONE

AN OBVIOUS VISION

THE SENSE OF WONDERFUL DEVELOPMENTS

TWO CULTURES FACE THE FUTURE

A BRIEF HISTORY OF THE XANADU CAPER

AN OBVIOUS VISION

AN OBVIOUS VISION

A computer is essentially a trained squirrel: acting on reflex, thoughtlessly running back and forth and storing away nuts until some other stimulus makes it do something else. A perfectly versatile enactor; by rigmaroles and enchantments we make the computer do our bidding.

But then what things should we have it Enact? How can it improve our lives? This is the important issue. That there is a technological imperative, some way it "has to" be done, is a myth and a fabrication. People get cowed, put in their place, when the technoids start enumerating the world as *they* see it.

TCMORROW'S WORLD OF SCREENS

Computer viewscreens can bring words and pictures right away. Businesses know it; there are perhaps 4 million computer screens now active in the country in business environments.

Individuals, unfortunately, just don't get it. Most, or "all," of our reading and writing can or will, in this century, be at instant-access screens. The question is not *can* we do everything on screens, but *when* will we, *how* will we, and how can we make it *great*?

To me this is an article of faith; its simple obviousness defies argument. If you don't get it there is no persuading you; if you get it you don't need to be persuaded.

What I don't understand is the apathy about this in the computer field. There is no sense of urgency; there is no unifying vision of uplift for humanity as soon as *every person gets a screen*.

TWO HOPES

1. We ought to have our everyday lives made simple and flexible by the computer as a personal information tool.

2. And we ought to be able to read and write on computer screens, with vast libraries easily, instantly and clearly available to us.

The woods are abuzz with supposedly great new computer services that will supposedly be offered to the public. Many computer people are "working in these areas." Yet what they give us time and again is complication, complication, because nobody has taught them how to *design simplicity*. "Not my job, man." No computer school teaches it. Beneath them, I suppose. Or more likely not imagined by them.

Simplicity does not come in pieces. You can't buy it in sections or add it in parts, on weekends. A thing is unified and clear and simple because it is designed that way, or it is not unified and clear and simple. Making things clear and simple is *hard*. (Biological unity is another matter-- it takes a long time and millions of mistakes, and does not necessarily act in our perceived interests. Don't pull that analogy.)

The starting point in designing a computer system must be the creation of the conceptual and psychological environment-- the seeming of the system-- what I and my associates call the *virtuality*. You begin there and work back into the mechanics. You decide how it *ought* to be, and then make that vision happen; you don't just patch and splice and add and adapt.

The two words that characterize life at computer screens are BINGO and OOPS-- Bingo because things come the instant you call them, and Oops when you did what you didn't intend-- which in bad computer systems, *most* computer systems, is hard to undo.

Pragmatism and the desire to get along in the world lead people to put up with what should not be put up with. But nothing really stops anyone from creating the good and the elegant except habit, inertia and desuetude-- and the fact that doing right is much harder than not doing right.

As soon as you understand computers, all this should become obvious. Yet most people have not understood computers-- partly because some computer people didn't want them to-- and so the benefits to our lives have been put off and put off.

Now laymen are getting their personal computers-- Apples and Radio Shacks and so on-- and independently seeing, many of them, how things might and ought to be. (Though the inexpensive computers are being called *micro*computers so that newcomers will somehow think they're different from the old ones, and subtly inferior. The word "microcomputer" leads people into thinking that the new dinky computers are in some way not as good. Thus the word is like the word *nigger*-- suggesting unspoken inferiorities without having to name any.)

HOPE 1. SIMPLIFYING OUR LIVES

Computers should bring simplification, rather than complication, to our lives: they should handle the minutiae, the snibbety details of day-to-day existence. Computer screens should bring us the everyday data of our lives -- whatever memoranda we use-- effortlessly so we no longer have to deal with myriad scribbles on paper. What you write down for your own use should be *always available from a screen*, not randomly lost and buried. Birthdays, appointments, possibilities to be kept track of, the blizzard of everyday natter,; the scheduling of our lives (which is very complicated in principle, and which we blunder through, sometimes with great difficulty); the trivia of bookkeeping (which most people make into a yearly chore in relation to the IRS); the cross-indexing and storage management of the things we keep (conventional wisdom says we should keep less-- actually a reflection, I believe, of the fact that our systems are lousy and therefore very inconvenient).

So we need *unified* personal systems for a variety of purposes, tying these objectives together. Now, most computer people are under the impression that this implies a vast amount of *programming*. I say no: what it requires is a lot of *good design*, and the creation of some very simple building blocks. (As a clue to the sophisticated reader, let me add that exactly two examples of such systems are known to me: they are called Visi-Calc and SDMS, but we have no time to go into them here.)

HOPE 2. ACCESS TO IDEAS

In a day of the "information explosion," with more and more being _printed_, the most up-to-date people use the telephone much more than the library.

The second hope I mentioned earlier was that we could read from and write on screens with new freedom. Those of us who grew up passionately believing in ideals that made our country great, such as liberty and pluralism and the accessibility of ideas, can hardly ignore the hope of such an opening-out. Libertarian ideals of accessibility and excitement that might unseat the video narcosis that now sits on our land like a fog; with alternative explanations so anyone can choose the pathway or approach that suits him; with ideas accessible and interesting to everyone, so that a new richness and freedom can come to the human experience; a rebirth of literacy; etc. All that is what this book is about.

Yet dammit, what's worst is everybody lacking a sense of urgency. This is the eleventh hour of the human race, man. There is a deadly urgency about everything we do.

LET'S DO IT

These two hopes-- the simplification of our lives, the cornucopia of ideas and writings and pictures-- are the focus of my own work. Twenty years ago, in graduate school, the two hopes I have mentioned came to me, as I hope they have come to you one way or another. I have put a lot of time into trying to make these things happen in ways I consider right, which I used to think were obvious to any idiot but apparently aren't.

In future writings I will deal further with the design of simplicity. (Meanwhile my two-part piece, "Interactive Systems and the Design of Virtuality," in the November and December 1980 issues of _Creative Computing_, is A START.

In the current volume, however, I will deal simply with reading and writing from screens, and the universe that I think is out there to create-- and then explore and live in. Vannevar Bush told us about it in 1945 and called it the memex ("As We May Think," _Atlantic Monthly_, July 1945, 101-8), but the idea has been dropped by most people. Too blye-sky. Too _simple_, perhaps.

This book presents a dream, a wild surmise that perhaps many have had but most have kicked under the bed as unfit for daytime contemplation. I have tried to capture it, a ludicrous butterfly-hunter scampering through unknown territory.

Okay, maybe it will turn out to be impossible. But if not the details, if not our valiant try, perhaps the *vision* may endure-- and perhaps also some readers may achieve an expanded realization of what it means to make simplicity happen.

I think this whole dream is possible. I may be wrong but I've given it my best shot, and here you are. This book describes it so that the reader-- anyway, the technical non-timorous reader-- can decide if it's something he *wants*. If not goodbye. If so, well -- join the club. We can be reached.

THE SENSE OF WONDERFUL DEVELOPMENTS

No alert person, drubbed by popular magazines and TV news, can fail to have heard that we are on the threshold of some sort of new era in the user of information. Soon, we hear, we will be able to get at the Library of Congress stored on a disk, or movies in a pinky ring, and information that we want vaguely may come at us without our even having to ask.

A hundred jarring systems are confuting. Many media moguls-- "smart money"-- think they have it all worked out; although in different directions. Corporations are being formed. The hearts of investors are palpitating. Foundations and federal agencies are continuing to put out money for breakthrough showcase projects. Yet, in my estimation, we have not a state of progress but a state of confusion. Never before have so many accepted the unrefined technical fantasies of the few. Never before has so much been spent for what has been so little understood or thought out.

Unfortunately, the public has no simple comprehension of the varieties of possibilities, the vast range of options. They will believe anything they are told except the whole picture, which nobody tells them. Laymen have no longer the slightest idea of what is going on. The gap widens continually.

This sort of thing happens easily in any field. Technical people create catchphrases, and people from outside, eager to be up-to-date, seize on the catchphrases as received wisdom, ideas that seem to span and comprehend all the possibilities. Those outsiders now spread the gospel to their own corners of the world, never quite sensing what an arbitrary selection has been made for them; failing to ask pointed questions, they in turn become opinion leaders for other outsiders who are even more afraid to ask. To mix parables, it is as if the blind men, after evaluating the elephant, then lead the other blind men in their several directions.

A variety of people are proposing arrangements by which other people, meaning we the public, should handle information in th future; and accordingly the public ought not submit with docility to just whatever may result spottily by chance. A phrase often heard is, "anything you want, instantly." On closer investigation, however, it turns out that there is much disagreement as to what you want, as well as considerable disagreement as to what instantly means.

(In my book Computer Lib I endeavored to sort out for laymen the differences between the fifty or so most important systems for retrieving and presenting information. That book has done rather well, but mainly among readers who already knew the subject.)

THE CABLE BABEL

Videocable operators think the public is ripe for about anything they offer. (Indeed, in England, France and other countries so-called "videotex" and "teletext" systems are already in operation, offering a variety of specialized information to the TV user, and their enthusiasts think it could revolutionize the world.)

THE OFFICE OF THE FUTURE

Souped-up "workstation" computers on one big cable, the Ethernet: that's the Xerox vision, supported by DEC and Intel-- mighty computer corporations.

HELLO CENTRAL, GIVE ME HEAVEN

Another view is held by companies that are selling office interconnect systems built around the telephone-- such as Rolm and Datapoint.

TEXT SYSTEMS

Computerized text communities are springing up. Offices find they can tie their "word processors" together, speeding information between executives. Time-sharing systems offer "electronic mail," which has the advantage of not interrupting the recipient (as does the telephone), but still getting a lot communicated fast.

A crucial development, the Arpanet connects university and military computers all over the country. It turns out that its main use, though, is sending messages among its users.

All this has led the post office to jump into "electronic mail" with both feet; its new ECOM service will begin next year.

Computer Bulletin Boards, "Community Information Systems," teleconferencing systems, all are creating new communities that share text via computers. The armed forces, too, have complex text systems (where commanders can read all messages of those below them, but not vice versa).

ELECTRONIC PUBLISHING

Electronic publishing is coming, this much all agree on. Just what it will be is not so clear.

For some five hundred years the educated public has been reading from books and magazines of paper.

Now all of that may change.

As screens become more and more available, there is less and less reason for printing on paper. The costs of wood pulp and gasoline, the long lead times of paper editorship and production, the increasing divergence of specialized interests, the lowering cost of computers with screens, of disk storage and digital communications, all suggest this.

Beginning thinkers in this area often suppose that what will be offered to the screen reader will be merely individual stored documents, available on line quickly, but based somehow on conventional documents enstling in conventional computer files.

Our point of view is different.

Many approaches to electronic publishing are very complicated. But that can't work on a broad scale; "publishing" suggests use by the public. Meaning simplicity.

WHITHER?

All these approaches are different. They seem to be converging, but are they? They do not combine well; hooking them together creates something like the New York subway system.

The point of view I would like to suggest is that we need unified design. It has to be simple. It has to be powerful. It doesn't have to be complicated. In fact, can't be complicated. And perhaps it can be built from the "document" as we have long known it, the "author" as we have long known him, and an extended form of "writing" as we have long done it and read it, rather than what some people, such as McLuhan and the video freaks and the CAI folk, have been telling us would be anonymous, colle-tive, scrambled, psychometric, and/or Boolean. I believe there exists a clean, complete and thorough solution.

TWO CULTURES FACE THE FUTURE

C.P. Snow pointed out long ago that there are two educated cultures, the culture of technology and the culture of the humanities, and they don't talk to each other. That was twenty years ago, but it's still true.

Not only is it still true, but the two cultures have united on a *false*, *agreed-upon definition of what computers are*. In this polite conspiracy the members of the two cultures, technical and literary-- who rarely talk to each other -- have it all figured out.

Their false notion of computers is that they are Inhuman, Oppressive, Cold, Relentless; and that the somehow Reduce Everything to Mathematics.

One camp says "yessir, and I run 'em," and the other camp says, "I want no part of it."

Never mind wat computers *really* are. This view, in its two variations, is a strange fact of our culture and psychology. But it has virtually nothing to do with computers.

To throw things in a sharper light, let me refer to those with technical training as the *Technoids* (or Noids for short), and I will refer to those with a humanistic background, in literature, history, the arts, etc., as the *Fluffies*.

THE NOIDS

The technoics have an exaggerated and caricatured notion of what constitutes clear-minded thinking, and never miss a chance to denounce other cognitive styles as "illogical." Or to denounce people who have difficulty learning the complicated systems they, the technoids, dream up.

My favorite example is the typical technoid insistence that you can't type a number into a computer using the letter Oh, you have to use the numeral Zero, because otherwise it isn't Logical. This despite the fact that a computer can easily be programmed to recognize that when you type Oh in the middle of a number you mean Zero, just the way a program can distinguish between a decimal point and a period, or a hyphen and a minus-- contextually. But that's not the myth. For some reason a rigid and punitive notion of "logic" is important to such people.

NOIDS' OUTLOOK

The technoids are usually hired guns, interested in the next complex problem they can get into. They generally have an obsession with favorite methods, and a negligible concern for history, art, literature or human freedom. Indeed, some of them *like* to oppress (and some of this type get to head computer centers eventually).

In a famous experiment, psychologist Stanley Milgram, wearing a white coat, instructed unsuspecting subjects, who thought they were merely paid assistants, to push buttons that would inflict terrible pain on others. To Milgram's chagrin, nearly everyone followed instructions without a qualm.

This in a way characterizes the Technoid mentality. If the government solicits bids on a Deterrent Weapons System that will selectively barbecue only the small children of an Aggressor Nation, the technoid will probably say Yes Sir, Can Do, What Color Do You Want the Corpses? While the Fluffy who has read Sophocles and/or Tocqueville may be *slightly* more likely to say, Wait a Minute...

THE FLUFFIES

The Fluffy cognitive style leans toward vagueness and the reduction of issues to vague idealistic terms (they being unused to specifics except for Metaphors and Object Correlatives.) Their disposition is always to get away from specifics as being mundane and/or Sociological.

And they do not like computers or the idea of screens. "I love books," "I hate computers," "It sounds so cold," "I can't take it on the train (in bed, in a hammock, into the woods)" etc. "I can't see cuddling up with a CRT in bed." They have no conception of the importance inad immediacy of creating an electronic literature that reflects their own values.

I have experienced many levels of Fluffy negativism to computer ideas, which I call (simply as a measuring-stick), Fluffy-Indifferent, Fluffy-Resistive, Fluffy-Hostile and Fluffy-Aggressive. WE NEED NOT GET FURTHER INTO THIS AT THE PRESENT TIME.

LITTLE CORNERS

About the only thing the groups have in common is their shared view of computers. Their views of each other are mutually derogatory, roughly on the level of "Youre the one who eats weird food, not me!"

But one interesting aspect of the two cultures is their view of each other in the world. Each sees the other group as "those people in their little corner, unaware of the big wide world."

To the Fluffies this real world is history, art, literature, and the little corner is "technical things." To the Technoids the real world is that of Technical Questions and Ideas, and the little corner is the artsy-craftsy nook of bygone concerns.

SYSTEMS HUMANISTS

As you may have suspected, I see another point of view. As far as I am concerned both the Technoids and the Fluffies are in their own little corners. In the broader view, the goals are the long ones of civilization-- education, understanding, the preservation of human values-- but we must use today's technologies. I call this view systems humanism.

Civilization as we now know it is based in part on running water. That system had to be thought out. Similarly, somebody's gotta design waterworks for the mind. But it should be someone who understands the fluidity of thought.

A BRIEF HISTORY OF THE XANADU CAPER

Eagles don't flock.
You have to find them one by one.

H. Ross Perot

Expeditions and projects do not end in the spirit they began in. This thing began with long walks in Cambridge and has become different sorts of ordeal for many people.

Complicated ideas evolve slowly. People who do not work with ideas a great deal may not always realize the many steps, guesses, postulations, reconsideration and general mucking about such things entail.

In retrospect it is baffling. I know now that there was no reason to expect to find technical secrets-- they just happened to be where no one else was looking; there was no reason to expect a collection of eccentric geniuses devotedly to work the thing through without salary, but they did; there was no reason to expect we could advance to this point while retaining the majority ownership that assures it will be done right. But we have. Strange forces are at work, and we will try to stay tuned.

Of course, anyone else would have found the same things, if only he had looked in all the right places. But from the literature it does not seem anyone has.

Through all of it we applied a relentless pressure for consistency and simplicity, and the thing cooked down remarkably. The amazing fact was that it has worked, that the hard technicalities could be pushed to fit soft ideals. But only by intricate search. This could not have been done with schedules and deadlines. When a project requires both exhaustive exploration and unusual inspiration, it is going to take however long it takes.

OBLIGATORY "ON THE SHOULDERS OF GIANTS" PARAGRAPH

I far prefer to work alone on complex ideas, without other people's diversionary, obtrusive and irrelevant suggestions. It would never have occurred to me that I would have ended up assembling and shepherding a group of people smarter than I who would actually get the thing done, and get it done right, from time to time overruling my own ideas.

The project has from the first been carried out in a conspiratorial atmosphere on the assumption that I (later we) understood something others did not understand, and reached for ideals others were not yet ready to comprehend.

It would have been nice to get advice from Jefferson and DaVinci and some other heavies. (Just a "nice going" now and then would have helped a lot.) Unfortunately they weren't around, so we had to wing it. And the scholars and "humanists" who consider themselves the anointed heirs and guardians of their tradition weren't available for comment. So we have been thrown on our own resources.

WE WANT IT, YOU SEE

This thing has been created because I, and later my collaborators, wanted to use it. I have been sharpening a very big pencil for twenty years, and hope to live long enough to finish the other writing I have been accumulating so long in note form.

It may be of interest that we who made it want to be ordinary users of the system. The group is not, as a rule, particularly modest or retiring; the fact that we have created a system on which we desire no more than to be ordinary users should be taken to indicate, not that our wants are modest, which they are not, but rather that we want to put an emperor's resources at the fingertips of all users, especially children and scientists and poets. ("At last I can live like a human being!"-- Nero, on completion of his palace; "There need be no titles of nobility, since there will be no higher honor than to be called Citizen."-- Constitution or Federalist papers someplace.)

ALONE

I hated school all my life, from first grade through high school, unrelentingly and every minute. I have never known anyone who hated school as much as I did, although my assumption is that other dropouts do.

I dropped out of school in the seventh grade after one altercation too many with a certain teacher; sent from the room I kept on going. That was the basic dividing-line of my life. The sense of defiance, of finality, of burning my bridges, of contempt for the sheep who put up with it all, are with me yet. Cajoled back after a year out, I found nothing had improved.

My real education took place in bookstores-- to which I would sneak when avoiding the hated sports; in movie theaters (I lived where you could see a lot of English movies), from conversations with my grandparents and great-grandparents. And from magazines.

When I was twelve, my heroes were Bucky Fuller, Bertrand Russell and H.L. Mencken, as I recall. Also Walt Disney, of course, and Orson Welles.

A big moment in my life was when my grandfather took me to an exhibit of DaVinci's inventions. It was in the IBM building on 57th Street. I thought, "Wow!" What a great company IBM must be, to back a guy like <u>this</u>!" I was young; perhaps my expectations of IBM from that experience were too high. I always had a low threshold of indignation.

College turned out to be not like "school" at all. At last-- people were interested in <u>ideas</u>, in <u>talking</u>.

I did Philosophy, the gentleman's major, and explored things. Doing things from scratch, ignoring the way things had previously been done-- that was my thing; trying to start from first principles with unclouded vision. (There is a moral issue here. It is as wrong to do things because others don't do them as it is wrong to do things because others do. What should be done cannot be ascertained from watching other people.) I got out of college expecting to find new things to be done that others hadn't. I was proud of my own powers of abstraction and conjecture.

COMPUTERS

My second year in graduate school, fall 1960, I took a course in computer programming. The instructor, Arthur S. Couch, was an easy-going and reasonably well-informed guy. One of the things he told us right off the bat was that integrated circuits would make computers small and cheap very soon, even though the only computer at Harvard was the 7090 up at the Smithsonian Observatory-- a big machine indeed.

I will always be grateful that the course did not cover Fortran, and that instead we got into machine language and assembler-- the Real Stuff that let you see what really happened in the machine.

I announced as my term project a writing system for the 7090: the idea was to store your manuscripts in the computer, change them with various editorial operations, and print them out. (When this became commercialized years later, it was called "word processing." I called it, at the time, "text handling," which still seems to me the more appropriate term: you do not *process* words, you simply put them away and get them back out.)

My specs for the term project went much farther, however; the obvious notion of being able to revise text seemed to me obvious and inconsequential; the really *important* features would have to do with the organizing of the text for the user's benefit. So I designed in the additional features of alternative versions, historical backtrack, and revision by outline. (I'm not sure presently whether "links" were in the first specs or not)

It seemed so simple and clear to me then. It still does. But like many beginning computerists, I mistook a clear view for a short distance.

While it was obvious that *interactive* computers-- "one person, one computer"-- would be along in a year or so, tat system had to be configured around input commands punched on IBM cards. I had a dumb little input language worked out.

The long and the short of it is that I wrote thousands of lines of 7090 assembler code-- I think in the range of 40,000, but I'm not sure.

The project was not finished. I got an incomplete. Since Couch was such a good fellow, I suppose I might've found some way to make up the course, but I never did, which is one reason I only got a master's.

1/18

CAI

There was a lot of talk around Cambridge then about Computer-Assisted Instruction, for which there was a lot of money. I was originally all for it. After talking with CAI people, though, and rather soon after getting into the subject, my original editing ideas were expanded, and became what I first called the "thousand theories program"-- an explorable CAI "program" that would let you learn many different theories on many different subjects, at your choice. This rather quickly swung into what I would eventually call "hypertext"-- non-sequential forms of writing involving links. This was essentially my second design.

My other studies meant nothing to me now. I wanted to be in Computers-- but at that time there prevailed the silly notion that computers were "mathematical," and being a mathematical incompetent I was unable to get a job in the field.

THE PORPOISE WORKS

I did manage to find other interesting work, however, and went down to Miami for a year at John Lilly's dolphin lab. There I fed and petted the dolphins, got splashed by the notorious Elvar, and did photography and movie editing for Lilly's enterprise.

Lilly had a computer-- one of the original classic LINC computers developed by Wes Clark-- but that was somebody else's department, and I got no chance to use it, though it would have been ideal for what I wanted to do.

Still looking for a way to break into computers, and convinced that this revolution was right around the corner, but mystified at everybody's general obtuseness as to where it was going, I took a job teaching sociology at Vassar-- partly because I now had a family.

Now I was thinking about a third system, combining the old text editing and idea-management stuff with non-sequential writing. So a third system gradually became clear at Vassar. My notion was that of having sequences which could be linked together sideways; zippered lists, I called them.

The Vassar job provided time to start writing articles. And, to my amazement, I found it easy to get articles published.

PUBLISHING ARTICLES

My first paper was accepted by the ACM national conference, 1965. It was in Cleveland, but even so, I was thrilled. The audience was some 800 strong, and they really liked my presentation. (Little did I know it would be downhill from there.) I was briefly lionized in the field. I got invited to a VIP-researcher wingding at Lincoln Labs.

Then people lost interest. I was talking about the structure of ideas, and thus how to set computers up to hold them. Nobody got it. Everybody was listening for something else. Some people didn't want to see what I was saying, calling it "blue-sky" and "arm-waving"-- in other words, they basically lacked the capacity to visualize. If I had understood years ago how little capacity most people have for visualizing things, I'm sure I would have gone about matters rather differently-- though I can't quite see how.

I've also learned that most people are afraid of (and/or angered by) new words. Unless a thing comes on just the right silver platter, people don't want to think about it.

Since that time, always thinking it would help, I have wasted an unconscionable amount of time writing articles and giving speeches. I have come to learn that people's viewpoints are so entrenched, and their abilities to listen generally so negligible, that it is just not worth it. And most people aren't interested in ideas.

PUBLISHING HOUSE

My next job was at a large book firm. I wrote to its head; he was somewhat taken with my ideas and hired me; I reported directly to him. I have found then and since that the people at the top listen and understand better. This man was the brighest I have ever met.

He almost backed the Xanadu project-- it would have been at an opening budget of a qurter of a million-- but then they decided to go into CAI instead. I demurred, expressing my views of CAI, and we parted company.

This is when my designs got the name "Xanadu," it being a traditional name for a magic place of literary memory.

At that time my main design took the form in my mind of a sort of super Executive's Console, self-contained. But the idea of communicating between such devices was beginning to get through to me.

STRETCHTEXT

One of the hypertext designs I worked on at that time I called "Stretchtext." It was a stretchable form of text with a pyramidal structure. As you explored it got longer and shorter.

It was designed for the writing (and exploring) of history. It was actually designed around a Tektronix display, which is pretty hard to zoom on. (Zooming ideas have recently become popular, especially with the SDMS work of the Architecture Machine Group.)

SPOOKS

A government Intelligence Agency contacted me. I was so flattered at first, and so sure they must be very smart to have gotten in touch with me, that it took a long time to realize what chowderheads they actually were. They led me on, wasted my time, and had me form a corporation on promise of big contracts. No dice. However, some of the ideas I presented to them finally got through, as is evident from a recent book about their new system.

RING BUFFERS

In the 1968-70 era my concern was the organization of streams of data babbling through core memory, especially for use with "calligraphic" displays. I did clever stuff but it later turned out not to be the heart of the problem.

I spent about a year working on military text systems for a big lab. No ideas resulted.

A CERTAIN UNIVERSITY

About that time I wasted thousands of hours and thousands of dollars of my own money commuting to work with a group at a well-known university, but it turned out not to have been worth it.

They wanted to create a hot text system. My attempts to communicate the sorts of non-trivial structure I though were necessary for that system were dismissed as "raving" and "flaming." That there was a more sweeping outlook in my ideas beyond what they were choosing to implement was not considered by them as a possibility. They took rather the position that they had extracted the small gleaming insight in a morass of absurdity.

I emphasized to the system's developers that the word hypertext, as I had already defined it in print, properly referred to non-sequential writing, and that the interactive system itself should not be referred to as "hypertext."

To no avail: they consistently muddied up the terminology and referred to the damned thing as "Hypertext," with a capital H, to the galling effect that it has been referred to widely, since, as "IBM's Hypertext," with the capital H. Having been released through the IBM program library, it is available through IBM, but hypertext it certainly isn't.

I broke off relations with them not long after being required to sign a groveling paper saying that the "ideas" in the system were half theirs. I suspect the system has been widely influential in the development of "word processing." but who cares.

THE XANADOERS

By this time an individual named Jonathan Fagin invested some money in what I was doing. This brought a sense of movement; I recruited two others who moved the work on considerably. These were John Ridgway, then a sophomore at Swarthmore, and Cal Daniels, who then worked at Minicomputer Systems, Inc., and who had written their cassette tape operating system.

JOHN RIDGWAY

A clever second-generation Swarthmore student with hair down to his shoulderblades,-- incidentally, the first second-generation programmer I ever met, they were rare in those days, John Ridgway was an 1130 Fortran and folk-dancing whiz. And a very enjoyable guy. Naturally we implemented in 1130 Fortran.

CAL DANIELS

Soft-spoken, warmhearted, quietly clever. He lived in a section of Queens that looked like Old English houses, but the bachelor's interior of his own was startling in orange and tiger upholstery.

Cal was black, so-called, which does or does not deserve mention. The neighborhood kids teemed in and out of his livingroom to play chess.

Good meals, long evenings of discussion. I would explain something and Cal would stroke his chin and bob his head and say "Mmm..." And it turned out he saw problems far past where I was looking.

Cal's death in 1978 was a sorrow to all of us.

DISCOVERY OF THE ENFILADE

So the years 1971 and 1972 were essentially devoted to the problem of disk management and editing-- which turned out to mean Data Structure and fast editing methods that would always be up to date.

Gas was cheap then. I zoomed a lot between my Manhattan apartment, driving back and forth and back among Ridgway at Swarthmore, Daniels on Long Island and the R.E.S.I.S.T.O.R.S. kids' computer club in Princeton, talking the system, hashing details.

Anyhow, somehow we discovered the system we now call the first enfilade-- the Model T-- the data structure that manages huge agglomerates of text and their arrangement in and out of core, and which with its attendant routines edits fast and clean. Still secret, unfortunately, it is the granddaddy of the other enfilades which constitute our system.

Records of the discovery are spotty. However, it is clear that the Model T enfilade was fully formed by 6 March 1972, according to my design notes of that date.

Anyway, appropriate credit will be apportioned later when we can all sit down and figure out what really happened. Certainly it is the case that the help and advice from the Resistors (especially Nat Kuhn and Glen Babecki), and the detailed analyses of John and Cal, meant a great deal. I also vaguely recall jumping up and down and whooping with John Ridgway (in Parrish 22 at Swarthmore College) when we had discovered something incredible and codified it on the blackboard, but on combing hundreds of pages of notes that moment does not jump out at me.

In July 1972 the "Calgol" version was completed-- Cal Daniels' version of enfilade editing written in Algol-- but we had given the Nova back and had no chance to try to run it. (Too bad, because we could've had a fine, cheap word processor easily.)

Meanwhile John Ridgway continued with an interpretive version in 1130 Fortran. It eventually ran-- and actually drew a picture using its interpretive screen-language (which we called DINGO)-- on a Calcomp, in September 1973.

CIRCLE CAMPUS

In '73 they brought me to the University of Illinois in Chicago. A few weeks there made it clear I woudln't fit in with their computer establishment, so I contrived to write Computer Lib. I published the book myself in, I believe, August of 1974. It was an instantaneous success. Hundreds of orders came in. (At this writing some 40,000 have been sold.) It didn't make a lot of money-- clumsy business arrangements-- but its hidden invitations were to bring in the guys who would finally finish the work.

BILL BARUS

At about this time William F. Barus, whom I had known long before as a kid, took an interest in the Xanadu work.

Bill Barus was a graduate student in philosophy, brilliant, famously incomprehensible, with the unworldly kindness and deep moralism of a Li'l Abner.

It took Bill perhaps six months in the mid-seventies to work through with me the design of the system up to that date, since he was not satisfied merely to understand it; he needed to understand the theory behind each decision, the frame of mind, and the possible alternatives that might have been overlooked.

Then he thought about it for another six months, and tried out many conceptual alternatives. (We were sure, though, that we were going to create the system in PDP-11 machine language, using 16-bit codes for everything.)

My precious system up to that point was good on text storage but bad on links Bill announced a solution that would fix that. His new method would allow linkages to keep up with all changes. It was instaneous and permanent and could grow indefinitely. All changes, once made, left the file in canonical order, which was an internal rule of the system.

I did not understand Bill's solution for a long time. Like many others who have encountered his remarkable mind, I sometimes had great difficulty in following his ideas; so understanding it took another six months.

Until Barus' remarkable discovery, what we could do was essentially what anybody could do-- fast lookup that did not degrade too badly for large files. But Barus's work, which we refer to jocundly as "the eye in the pyramid," made possible an efficiently ever-linkable enfilade, a whole universe of poly-enfiladic structures. While his particular designs have been superseded, his stunning·insights opened the way to the world of unlimited linkages we believe we have found.

By now the personal computer field had opened up, and Barus and I tried several business ventures "to support the Xanadu work." But as in most cases of doing B to support A, A quickly got swept aside as we got swamped.

On an off chance I got a brief teaching appointment at Swarthmore, a chance to teach my own stuff. It didn't work out well. It proved more difficult to teach my own stuff, not easier. But trying to say certain things clearly for the first time did help my thinking quite a bit.

THE FINAL IMPLEMENTATION SQUAD

During 1978 a group of accomplices finally coalesced for the final assault; pledging lives, fortunes and sacred honor, and mostly a whole lot of time.

We are devoted capitalists all-- I from hatred of committees, blunted creativity and the dilution of thought; they from desire for their own space shuttle. Virtually all of us had awful school experiences; the fire that has driven this had to do with hopes for real change and liberation of the mind. Not your everyday people. Bright guys who had arotten time in school. Spacers, two of us, anxious to get off the planet immediately. And propelled by knowing we were onto something.

MARK MILLER

Mark Miller, at that time a Yale undergraduate and fan of Computer Lib, lived in Philadelphia. We talked virtuality for a while, then the Xanadu system. A superb programmer and remarkable theoretician with an infections smile and ingratiating waddle, he makes everybody happy; this despite his constant complaints about ther being too much gravity, and continual demand for the immediate abolition of all governments.

STUART GREENE

Stuart Greene had taken both my courses at Swarthmore, but went on to get a film degree at NYU. Devastatingly clever and probingly elfin, Stuart was teaching holography while he was in highschool and frequently goes on Buddhist retreats. In all things requiring manual coordination he is astounding, supposedly due to Zen meditation.

ROGER GREGORY

Roger Gregory defies description. Roger Gregory defies everything. Knowledgeable adn rancorously opinionated on all subjects, he defies you to argue by expressing himself as fiercely as possible. Miraculously, most people like him a lot, perhaps because he wastes no energy in hypocrisy. Some, however, would find him unkempt.

I first met Roger at some science-fiction convention in Chicago. He has gradually taken over day-to-day supervision of the project, cajoling, snarling, and demanding hotter spices.

ERIC HILL

While it was alleged that at fifteen he was involved in misdeeds involving a government computer over the phone, Eric was chortled out of juvenile court by an amused judge. In high school he was active in DECMUG (Digital Equipment Corporation MisUsers Group). He knows Systems. He is also more suave and worldly than the rest of us now that he is out of highschool.

ROLAND KING

Incredibly gaunt and thin, gentle and soft-spoken, Roland was a graduate student in formal linguistics when he joined us. The vanDyke beard suggests a Robin Hood, but underneath burns perhaps a fiercer Libertarian than all the rest of us. With his soft Southern accent, ever-present cigarette, faraway look and incredible graciousness, he is that member of the group on whom the term _gentleman_ would sit most comfortably.

THE AUTHOR

Oh yeah, and myself, glib, eager, sloppy, impatient, always behind. For 1980, during much of the group's work, I edited _Creative Computing_, which was the wrong way to spend the time but an interesting experience. It also permitted a little spreading of the word.

TWO YEARS BEFORE THE VAST

In the summer of 1979 the group rented an idyllic tree-shaded house and we _designed_. No premature coders _we_. Ever probing and reformulating, the group redesigned Barus's linking enfilade system, designed the historical trace enfilade, formulated the general theory of envilades (that the work of Miller and Greene), and by the end of the summer got into actual programming. The language, of course, is Bell Labs' C.* Despite a constant lack of funds, we muddled through.

It was a very special time: _dolce far niente_ design sessions on the porch with blackboards, long evenings talking design while Stuart, with his back to us, conquered Apple Breakout with one ball over and over. A special experience was a number of afternoons we spent with the very wonderful John Mauchly, listening to his reminiscences.

The group has fiercely and relentlessly pushed for generality. Several of my pet ideas went out the window, though they hung onto the sill for a time. One of those was the "literary link" formulation, expounded in various of my articles since about 1975, which is too tricky to go into at the present time.

* Lifeboat Associates' BDS C under CP/M was God's gift to us-- well, actually, the gift of its creator, Leor Zolman. It's terrific.

CHAPTER TWO

PROPOSAL FOR A UNIVERSAL ELECTRONIC PUBLISHING SYSTEM AND ARCHIVE

2.1 AN ELECTRONIC LITERARY SYSTEM

2.2 WHAT IS LITERATURE?

2.3 A TRUE STORAGE SYSTEM FOR TEXT AND OTHER EVOLVING STRUCTURES

2.4 A LINKING SYSTEM FOR TEXT AND OTHER DATA

2.5 THE DOCUMENT CONVENTION

2.6 COMPOUND DOCUMENTS

2.7 ELECTRONIC PUBLISHING

MAKING THE LITERARY SYSTEM UNIVERSAL

2.8 DISTRIBUTION AND NETWORKING

2.9 VITAL ISSUES

CHAPTER 2

PROPOSAL FOR A UNIVERSAL ELECTRONIC PUBLISHING SYSTEM AND ARCHIVE

We are all agreed that your theory is crazy. The question which divides us is whether it is crazy enough to have a chance of being correct. My own feeling is that it is not crazy enough.

Niels Bohr
quoted in Kenneth Brower,
The Starship and the Canoe, 46

PLAN OF THIS CHAPTER

This chapter is in nine sections, which introduce and elaborate on a very particular and precise design and plan. This chapter, design and plan are the heart of this book, a crossroad through which you are asked to pass repeatedly.

Some readers, especially those who may not have given these matters any thought, may find this material tough sledding the first time through. Therefore a summary level has been provided. (The bigger type.) Stick to that the first time through, or if you're in a hurry later on to pass to another Chapter Three.

If this chapter is long and tedious to read, that is only because it strives for completeness. I am sure that a few years from now everything in it will be quickly divined by small children sitting at a console which enacts these principles.

2.1 AN ELECTRONIC LITERARY SYSTEM

Here is the right way to do something by computer: first figure out what you really want to do and think about, instead of staying bogged down in what you usually do and think about when you don't use a computer.

What will be described here is the way we think information should be handled. In the later sections of the chapter we will be describing the detailed idea of it, the conceptual structure or virtuality. This chapter is only about this idea, without technicalities. (The few "computer technicalities" are in footnotes.)

We also believe that we have carried out this design in a practical form, and that it will shortly exist as a functioning computer program with many uses. This belief will be proven or disproven in the fullness of time. Meanwhile, what is really being described is what we think we have created. Believing that this is the right virtuality, it is what we have implemented.

The footnotes contain a few kibitzing remarks to those interested in how we have done it.*

* The Project Xanadu group has for some time been developing software to do what is described here with no complications for the user. Our way of seeing the world, as described here, is reflected in many ways in our unusual data structure.

(continued at bottom of next page)

For instance, we are going to look at ways of dealing with text based on its "true" structure, if we can figure out what that is.

THE DESIGN

It is difficult to describe an interactive computer system so people can visualize it. Most people have not had practice in visualizing jumping and responding objects on the basis of abstract descriptions.

When we say also that we think this design is simple and basic, like the telephone, that may be hard for some readers to believe, considering that it takes so many words to describe it. Yet we think this design, once understood, is spare, parsimonious and clear. (And that a few years from now, small children will understand it immediately when they get a chance to play with it.)

The structure of documents and links to be described here is, for a computer system, unusually simple. This is all there is; we will describe it completely. We regard the simplicity of this design as its greatest virtue. The user has no direct contact with technicalities. The technicalities underneath are simply the means whereby certain exact and simple services are rapidly performed.

THE INTRINSIC STRUCTURE IS WHAT YOU SHOULD SEE

The structure a user sees should be the intrinsic structure of his material, and not (as in many "word processing" systems) some amalgam combining the material itself with some set

(continued from previous footnote)

To do efficiently what will be described here, we have had to overthrow all conventions and conventional assumptions about data handling and indexing, building from the bottom up a system that we think can grow indefinitely without choking on retrieval and transmission bottlenecks. We believe we have achieved this in our unique proprietary software.

We could only carry out this design with the help of certain technical developments which are for the present proprietary and secret. A number of radical discoveries in the field of computer indexing and retrieval render it possible to offer these services within seconds on configurations of present-day equipment, even, we believe, as the number of documents and service requests expands to astronomical figures. See "The Only Way It Could Work," Chapter 4.

And we are going to create a service that simply stores and sends back different excerpts from this "true" structure of text.

---

of obtrusive conventions under which it is stored.

What the thing *is*-- its natural structure to the user-- is what he should see and work on: nothing less, nothing more, nothing else. It is therefore the representation of this structure-- of whatever structure the user may be concerned with-- that should concern us. However, the complexities of what may be wanted can be staggering. So the problem is to create a general representation and storage system that will permit automatic storage of all structures a user might want to work on, and the faithful accounting of their development.

WORLD AND VIEWPANE: BACK AND FRONT ENDS

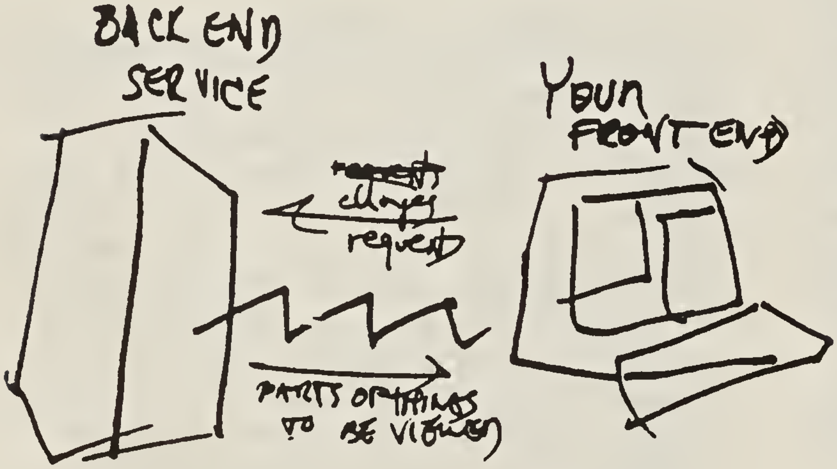
While computer display screens are to be the foci of our coming society, what the world is that will *show* on the screens is perhaps of greater concern.

The question in computerizing anything, then, should be *what is the true structure*? Having answered that, you design a system that stores and shows that true structure.

Storage is fundamental. What you store should be the basic structure of the information you are dealing with-- not some tricky arrangement that is carefully matched to some set of programs or hardware. (How you will look at this world when it is spreadeagled on your screen is your own business: you control it by your choice of screen hardware, by your choice ov viewing program, by what you do as you watch. But the structure of that world is the same from screen to screen.)

What we will discuss here is representation of the true structure of a certain kind of information, not how to show it. We believe that an orderly overall system can be developed for most types of written and graphical information and its instantaneous delivery.

That is the storage system we will be discussing here. Think of it as a "back-end" service that can supply your system. At the screen of your computer you explore what is stored, change it, add to it. The service we propose takes care of putting it away and sending you whatever part you ask for as fast as possible. That is the back end. What computer you watch it through, and how <u>that</u> machine is programmed, is your "front end"-- a separate problem.



This is a VIRTUALITY. One of the principles of designing virtuality is that there are many possible overall organizations that may be very similar. The problem of choosing among them is not simple. The tricky words that follow, "true structure," suggest utter uniqueness. But this is not the only "true" design. It is the design suggested by the one working precedent that we know of: literature.

Whether it is a good and right design is the central question for the reader to judge. What we describe can be done: if not by us, then by somebody sometime. But if it is not the right system, then it is on the reader's shoulders to come up with a better one.

Suggestons are welcome-- <u>if</u> you are sure you understand this design first (including Balance I. See "Tuning the System," a Chapter 3.)

2.2 WHAT IS LITERATURE?

Literature is an ongoing system of interconnecting documents.

THE LITERARY PARADIGM

A piece of writing-- say, a sheet of typed paper on the table-- looks alone and independent. This is quite misleading. Solitary it may be, but it is probably also part of a literature.

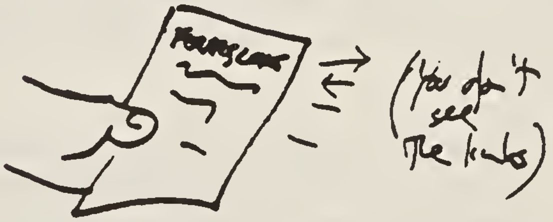
By "a literature" we do not mean anything necessarily to do with belles-lettres or leather-bound books. We mean it in the same broad sense of "the scientific literature," or that graduate-school question, "Have you looked at the literature?"

A literature is a system of interconnected writings. We do not offer this as our definition, but as a discovered fact. And almost all writing is part of some literature.

The way people write is based in large part on these interconnections.

A person reads an article. He says to himself, "Where have I seen something like that before? Oh, yes --" and the previous connection is brought mentally into play.

What is a document?



Consider how it works in science. A genetic theorist, say, reads current writings in the journals. These refer back, explicitly, to other writings; if he chooses to question the sources, or review their meaning, he is following links as he gets the books and journals and refers to them. He may correspond with colleagues, mentioning in his letters what he has read, and receiving replies suggesting that he read other things. (Again, the letters are implicitly connected to these other writings by implicit links.) Seeking to refresh his ideas, he goes back to Darwin, and also derives inspiration from other things he reads--the Bible, science fiction. These are linked to his work in his mind.

In his own writing he quotes and cites the things he has read. (Again, explicit links are being made.) Other readers, taking interest in his sources, read them (following the links).

In our Western cultural tradition, writings in principle remain continuously available-- both as recently quoted, and in their original inviolable incarnations-- in a great procession.

So far we have stressed some of the processes of referral and linkage. But also of great importance are controversy and disagreement and reevaluation.

Everyone agrues over the interpretation of former writings, even our geneticists. One author will cite (or link to) a passage in Darwin to prove Darwin thought one thing, anohter will find another passage to try to prove he thought another.

And views of a field, and the way a field's own past is viewed within it, change. A formerly forgotten researcher may come to light (like Mendel), or a highly respected researcher may be discredited (Cyril Burt). And so it goes, on and on. The past is continually changing-- or at least seems to be, as we view it.

There is no predicting the use future people will make of what is written. Any summary, any particular view, is exactly that: the perspective of a particular individual (or school of thought) at a particular time. We cannot know how things will be seen in the future. We can assume there will never be a final and definitive view of anything.

And yet this system functions.

LITERATURE IS DEBUGGED.

In other words, even though in every field there is an ever-changing flux of emphasis and perspective and distortion, and an ever-changing fashion in content and approach, the ongoing mechanism of written and published text furnishes a flexible vehicle for this change, continually adapting. Linkage structure between documents forms a flux of invisible threads and rubber bands that hold the thoughts together.

Linkage structure and its ramifications are surprisingly similar in the world of business.

A business letter will say, "In reply to your letter of the 13th..." Or a business form, another key communication, may say in effect, "In response to your order of the 24th of last month, we can supply only half of what you have asked for, but can fill the rest of the order with such-and-such item from our catalog." All of these citations may be thought of as cross-linkages among documents.

The point is clear, whether in science or business or belles lettres. Within bodies of writing, everywhere, there are linkages we tend not to see. The individual document, at hand, is what we deal with; we do not see the total linked collection of them all at once. But they are there, the documents not present as well as those that are, and the grand cat's-cradle among them all.

From this fundamental insight, we have endeavored to create a system for text editing and retrieval that will receive, and handle, and present, documents with links between them. We believe there is something very right about the existing system of literature; indeed we suspect that there are things right about it that we don't even know, as with Nature. And so we have tried to mirror, and replicate, and extend, existing literary structure as we have here described it.

2.3 A TRUE STORAGE SYSTEM FOR TEXT AND OTHER EVOLVING STRUCTURES

We are going to propose a way of keeping information that may seem odd and inefficient at first, but turns out to have remarkable power later on.

PROLOGUE:
MAKING EXTRA COPIES ALL THE TIME

In most computer applications (such as the layman's newest game, "word processing,") it is often necessary to keep repeated copies. This frequent and disagreeable problem has several purposes.

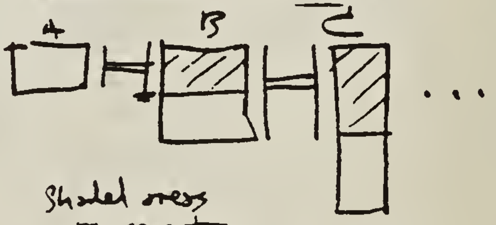
The obvious purpose, often thwarted, is to assure the safety of recent work against various kinds of accident. But that can be gradually ruled out: many systems are coming to make reliable "safety copies" automatically.

A more fundamental use is to keep track of former states of the work, in case mistakes or wrong decisions need to be undone. This need, <u>backtrack</u>, is serious and important. We do not need to go back through previous material often, but if we need to do it at all we ought to be able to do it rignt. Here is what doing it right entails.

DEPARTMENT OF REDUNDANCY DEPARTMENT

Note, however, that the conventional means of storage is rather silly. It involves making a complete copy of everything you've done so far. If what you're doing is making repeated small changes and additions, then you are repeatedly storing the same material, redundantly.

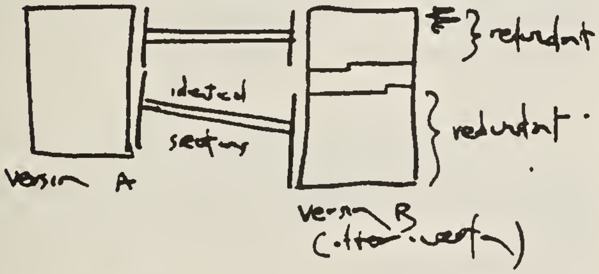
CONVENTIONAL BLOCK STORAGE



Shaded areas represent superfluous repeated storage of same material.

Under many circumstances the writer, or "text user," needs to reach back to a former condition. This is in the nature of creative work.

Virtually all of computerdom is built around this curious convention.* Most computer people will tell you that is the way God intended computers to be used.

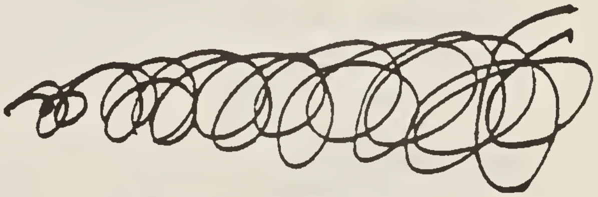


However, we can program computers any way we like, if we throw out the system software, and what we are talking about is creating new ways of doing things.

THE ALTERNATIVE

Suppose we create instead an automatic storage system that takes care of all changes and backtrack automatically. As a user makes changes, the changes go directly into the storage system; filed, as it were, chronologically.* Now with the proper sort of indexing scheme, the storage facility we've mentioned ought also to be able to deal with the problem of historical backtrack.

Think of it this way. An evolving document is really not just a block of text characters, Scrabble tm tiles all in a row; it is an ongoing changing flux. Think of its progress through time as a sort of braid or vortex. A DOCUMENT IS REALLY



AN EVOLVING
ONGOING BRAID

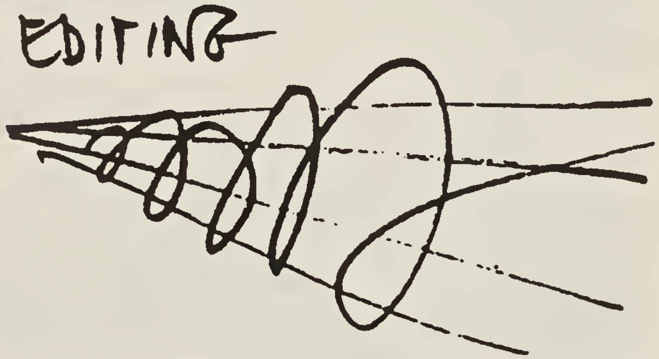
time →

* Including block-transfer circuitry, most display buffering, and the disk routines supplied with conventional operating systems.

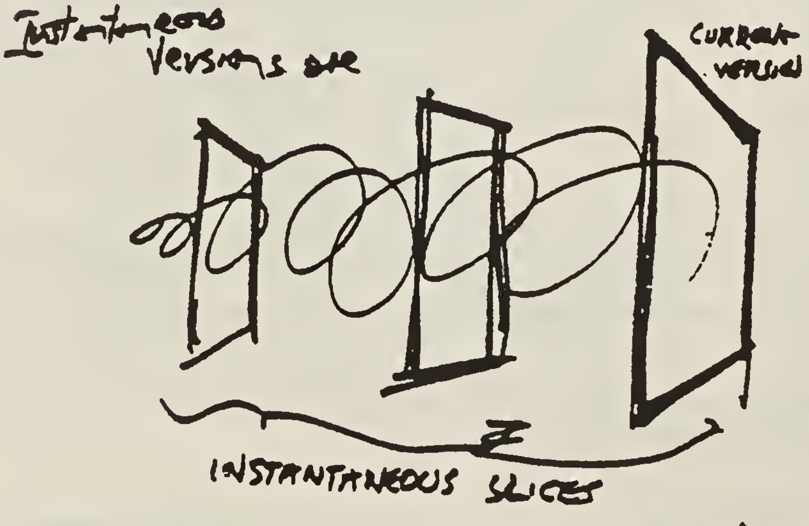
** (Of course, since the storage system assimilates all changes, it becomes nearly the whole "word processor," except for the user's front end.)

The true storage of text should be in a system that stores each change and fragment individually, assimilating each change as it arrives, but keeping the former changes; integrating them all by means of an indexing method that allows any previous instant to be reconstructed.

Think of the process of making editorial changes as re-twisting this braid when its parts are rearranged, added or subtracted,

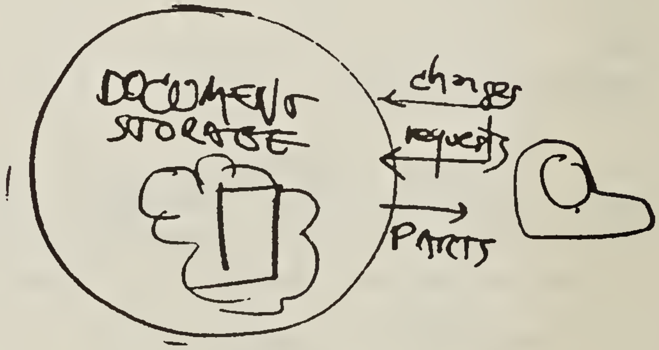


and think then of successive versions of the document, at successive instants of time, as slices in this space-time vortex.



Very well: the file management system we are talking about automatically keeps track of the changes and the pieces, so that when you ask for a given part of a given version at a given time, it comes to your screen.

The user may then refer not merely to the present version of the document; he or she may go back in time to any previous version. The user must also be able to follow a specific section back through time, and study its previous states.



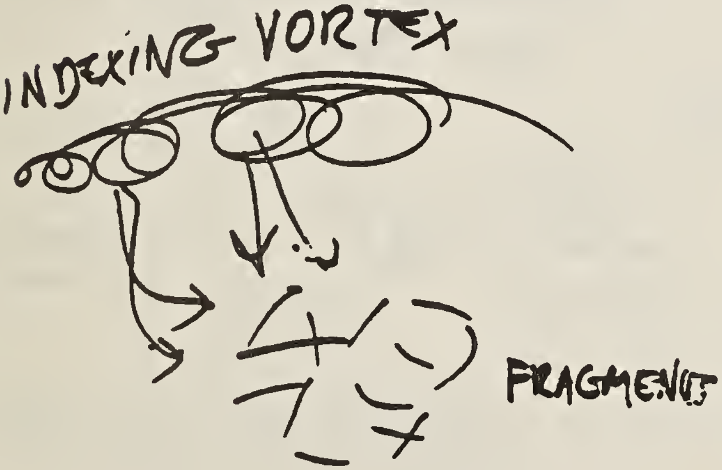
This can be done efficiently if the user is reading from a computer screen; since you can set up the system to reconstruct hastily any piece that is wanted at the instant it is wanted. THE PART YOU WANT COMES WHEN YOU ASK FOR IT.

PART-POUNCE

This system is built around the assumption that you are reading from a screen, not from paper. When you "go to a certain part" of a document, that whole document is not ready to show; yet the system gives you that part instantly, assembling it on the run from the many fragments of its actual storage.

We call this pounce. You pounce like a cat on a given thing, and it seems to be there, having been constructed while you are, as it were, in midair. Unlike things which dematerialize when you pounce on them, like cotton candy, this materializes when you pounce on it. I can think of no other example, except perhaps Potemkin villages.

* Obviously such a system departs from conventional "block" storage, and rather stores material in fragments under control of a master directory indexing by time,-- and other factors.

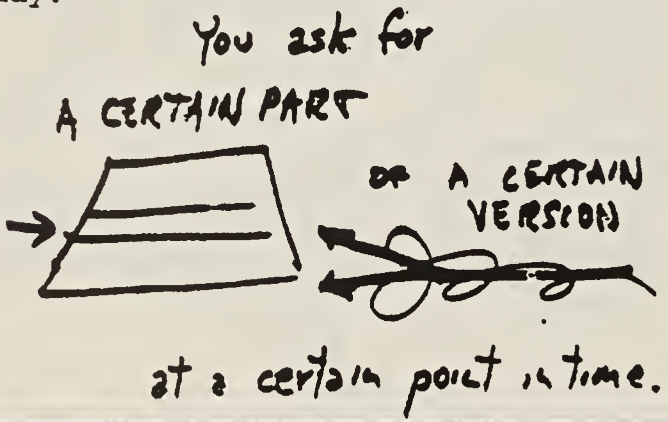


This method stores the document canonically as a system of evolving and alternative versions, instantly constructed as need from the stored fragments, pointers and lists of our unusual data structure. Thus there is no "main" version of a thing, just the ongoing accumulation of pieces and changes, some of whose permutations have names and special linkages. In other words, our system treats all versions of a documetn as views extracted from the same aggregated object. It will be readily apparent that the only way to do this is effectively to have direct track-and-sector access to the disk system.

(footnote continued on next page)

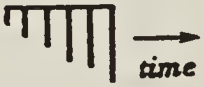
This is the true structure of text, because text is best viewed as an evolving, Protean structure.

You get the part you want next; the mistake of the conventional computer field has been to assume that the whole document had to be formed and ready.



ANOTHER VISUALIZATION

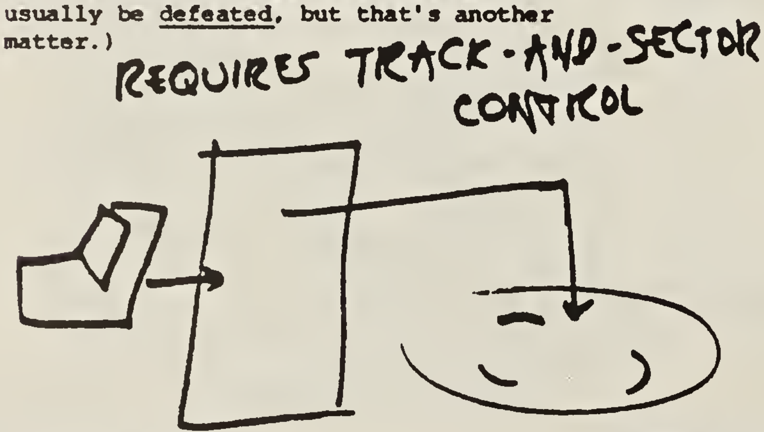
The canonical documents in this system can store teh same material in numerous different versions-- as, for example, in the successive drafts of a novel.

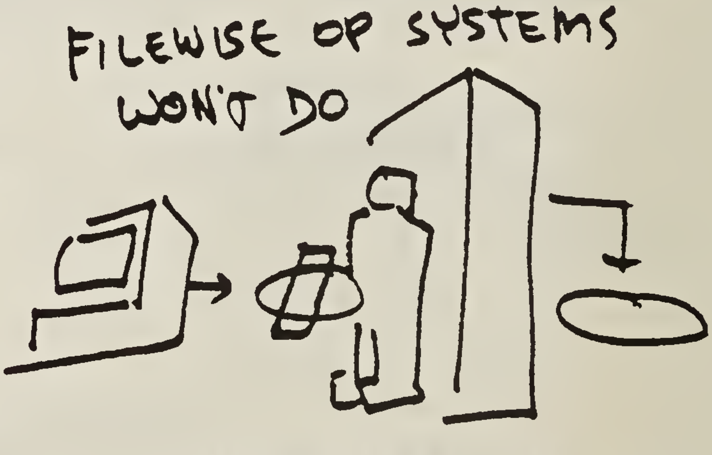


While the user of a customary editing or word processing system may scroll through an individual document, the user of this system may scroll in

(continued from previous footnote)

This is in contradistinction to conventional operating systems, from CP/M to OS/370, which typically deliver the whole file on every request. The illustration depicts the operating system as Frankenstein's monster with a silver tray. (These features may usually be defeated, but that's another matter.)

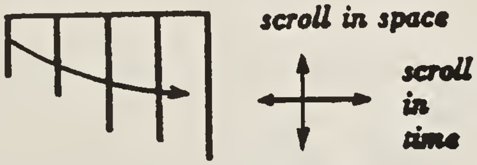




And storage space is saved by not having to keep redundant parts. This in itself is not very important.

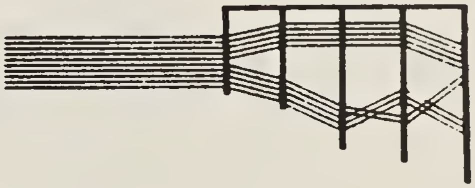
Such storage permits easy reconstruction of previous states for mental clarification, fresh starts, and transfusions of previous ideas. It also permits multiple uses of the same materials for alternative versions and "boilerplate."

time as well as space, watching the changes in a given passage as the system enacts its successive modifications.



Versions of a document set apart for other reasons-- "alternative" versions -- may likewise be flipped through or efficiently compared side by side.

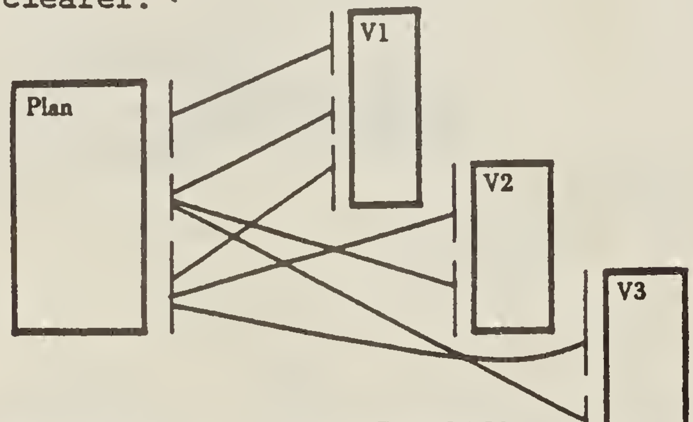
We call this system of storage Prismatic because we may think of a given part, or section, as being prismatically refracted when we pass from one version to another. We believe our Prismatic storage can support virtually instantaneous retrieval of any portion of any version (historical or alternative).



This same approach-- storage as an evolving structure with backtrack-- may be extended toall forms of data that are created by individuals.

ALTERNATIVE VERSIONS

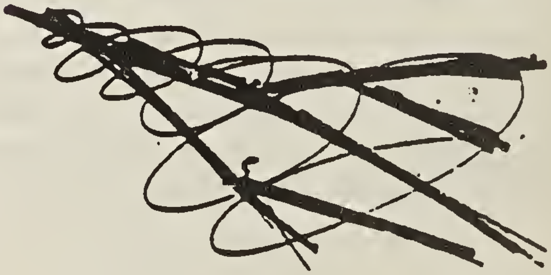
This same scheme can be expanded to allow alternative versions-- more than one arrangement of the same materials, a facility that writers and programmers would certainly use if it were readily available. Alternative versions (or Alts) are also important in many boilerplate applications, such as law and public relations, where the same materials are churned out repeatedly in different arrangements and variations. A master indexing scheme could greatly reduce storage requirements in these applications, as well as make the relations among the Alts much clearer.*



ALTERNATIVE VERSIONS

Actually, we may best visualize these alternative versions as a tree in the ongoing braid, a forking arrangement whereby one document becomes two, each of these daughter documents may in turn become others, etc.

TREE OF ALTERNATIVE VERSIONS



* By arranging for alternative versions to share common storage of the document's fragments, again we save space.

Pictures, and graphical data structures created at a screen, evolve in the same way and should be stored in the same way.

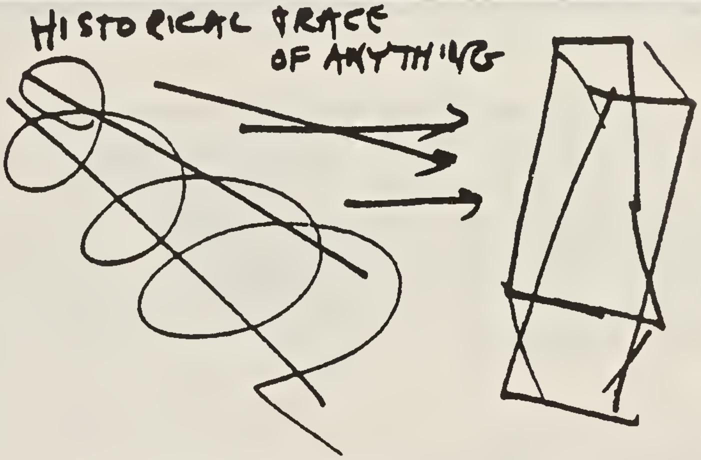
SAMENESS AND DIFFERENCE DISPLAY

Of course, a facility that holds multiple versions of the same material, and allows historical backtrack, is not terribly useful unless it can help you intercompare them in detail -- unless it can show you, word for word, what parts of two versions are the same.

Lawyers need this to compare wordings. Congressmen need this to compare different draft versions of bills. Authors need it to see what has happened to specific passages in their writings between drafts. Biologists and anatomists need it to compare corresponding parts of animals (assuming a graphical data base of physiology that shows evolving structure).

ANY FORMS OF DATA

This storage and indexing by pieces and changes works not merely for text; it can be used for any forms of data structure.



Thus if your are designing a building on a computer screen, as architects now do, you may browse through the changing design in the sequence you modified it over time, and create alternative versions as you like which share the common material.

Example.

An excellent airplane, the Boeing 747, now exists in a dozen or more versions that you may order from the factory. Complex blueprints exist for each of these versions, as well as lists of parts, etc. (Much of this is kept on computers as 3D data structures at Boeing, and perhaps quite well; this is simply a convenient example, and no criticism is intended.)

* (Such intercomparisons would in a more conventional system require writing and invoking search commands of some complexity among the various related files.)

As a first step we propose such an evolutionary structure, the docuplex, as the basic storage structure for electronic literature.

Using the data structure and programs we have described, it is possible to store all the 747 designs as one unified data pool, with the forking-version facility reading the variant designs directly out of this single structure.

SIMPLICITY ONE

By creating such a capable storage system, we have greatly simplified the life of the text user. The nuisance of backup (and the spurious nonsense-task of finding names for backup files) are eliminated. But more important, we have unified all versions (previous and alternative) in a unified structure, the docuplex, permitting part-pounce on present, past and variatn structures. The user may scroll through any two versions to see corresponding parts; and much more.

STAGE ONE ALL TOGETHER

I have so far presented several new capabilities that I think are important: alternative versions and historical backtrack, both with sameness display; and links.

These work together; they have to. The links allow the creation of nonsequential writings, bookmarks and jump-structured graphics of many kinds. But if you are going to have links you really need historical backtrack and alternative versions.

Why? Because if you make some links on Monday and go on making changes, perhaps on Wednesday you'd like to follow those links into the present version. They'd better still be attached to the right parts, even though the parts may have moved. And the sameness-display allows complex linked alternatives to be studied and intercompared in depth.

So let us call this Stage One: a system of computer storage that holds pieces of a thing, not big blocks, and assembles them instantly into whatever part of whatever version you ask for; that allows you to create links of any kind you want between any things you want; and shows you which parts are the same between related versions.

Let us call such a storage system a hyperfile.

You don't <u>have</u> to use these facilities. You can store text in long blocks if you wish. But if the facility is there, then the people who need it can use it.

Perhaps most important, these facilities provide a building-block for what is to be described in what follows.

2.4 A LINKING SYSTEM FOR TEXT AND OTHER DATA

Assuming that we are storing materials in such an evolutionary structure, the creation of "links" to the material becomes much easier.

LINKS ARE PART OF THE WRITING

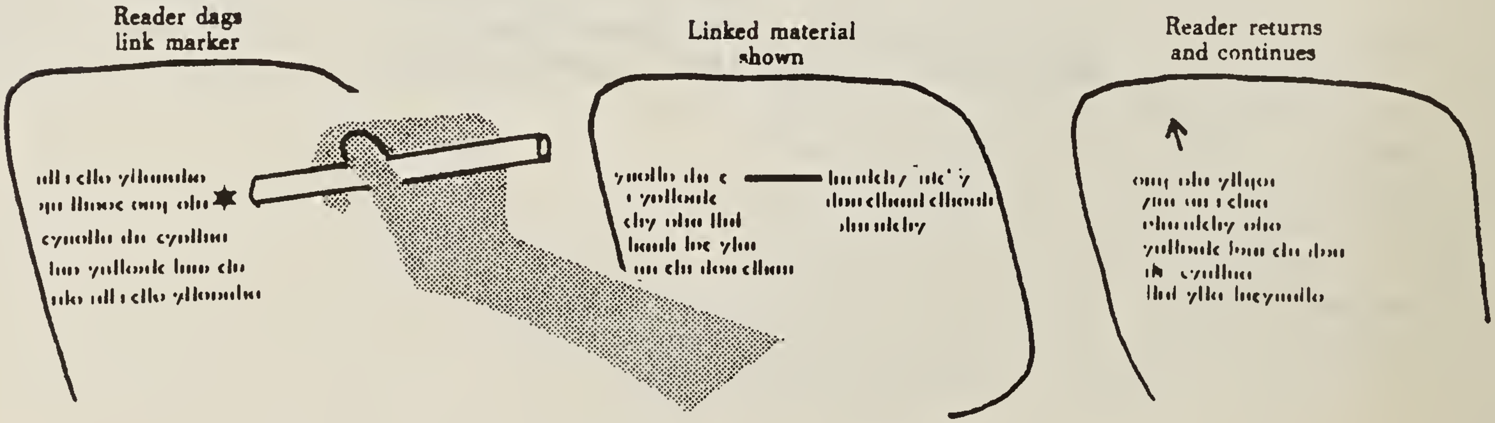
A link is simply a connection between parts of text or other material. It is put in by a human. Links are made by individuals as pathways for the reader's exploration; thus they are parts of the actual document, part of the writing.

As perhaps the simplest type of link, a user may create book-marks--places he may want to re-enter text when returning to it.

JUMP-LINK

As another simple first example, let us simply think of a link as some sort of a jump opportunity, like a conventional footnote. An asterisk, say, signals that "there's something to jump to from here." If you point at it with your lightpen (or mouse or whatever), Bingo!-- you're now at the footnote, or whatever else the author took you to. If you don't like it there, hit some sort of a Return Button and it pops your previous address from a stack, so here you are back where you were and no harm has been done.

JUMPING ON A LINK

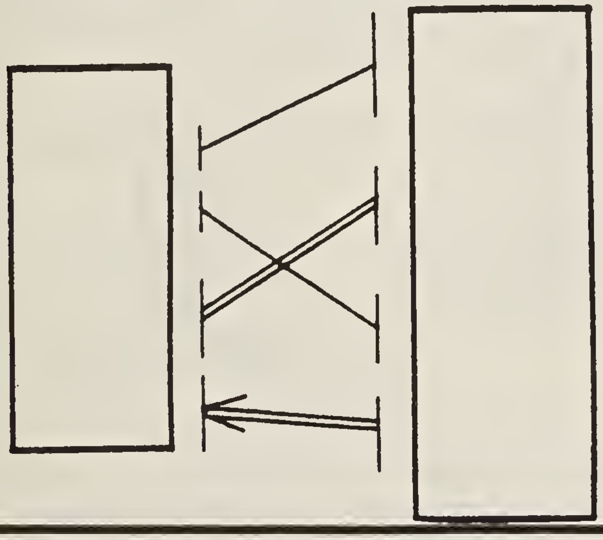


You may want links for commentaries, bookmarks and placemarkers, footnotes, marginal notes, hypertext jumps and innumerable other uses; but they are very hard to keep in place with conventional computer storage structure.

MARGINAL NOTES, SIDE-BY-SIDE WRITING

Marginal notes are another simple and important type of link. (Where the "margin" of the screen is-- that is, how to show them-- is a matter particular to your own screen setup.)

PARALLEL LINKED TEXT



A user may also make side-by-side connections of other types. On contemplating any two pieces of text, he may make a link between them. Thereafter, when he displays either piece of text, and asks to see the links, a link-symbol is displayed, and the other attached text-- if he wishes to see it.

Naturally, making a marginal note consists of writing the note and hooking the link.

HYPERTEXT

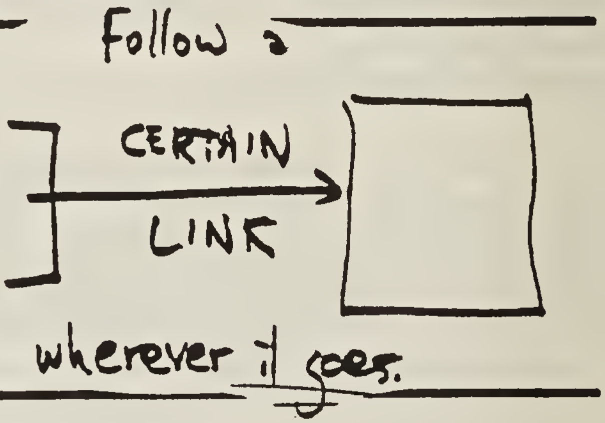
The link facility gives us much more than the attachment of mere odds and ends. It permits fully non-sequential writing, or hypertext.

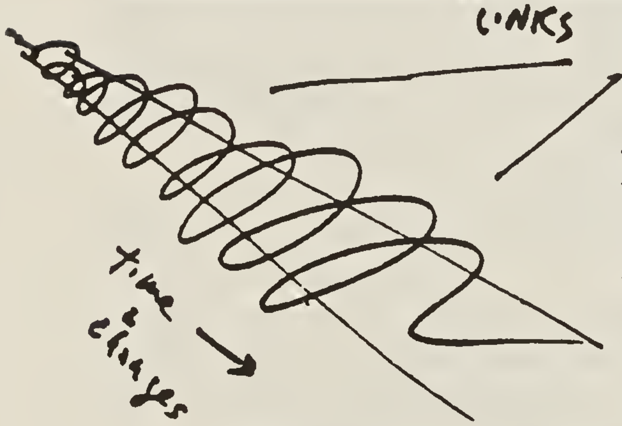
* LINKS + PRISMATICS → USABILITY

Most computer schemes for linkage face the terrible problem of "updating" the links as text is modified and successive versions come into being. The present scheme dodges this problem smartly (at least at the local level): a link is attached, not to a positional address in a given version, but to specific characters, and simply stays with them wherever they go. Thus Prismatic storage solves a considerable problem.

However, the evolutionary storage we have already described allows any links to be associated firmly with the pieces of data in any evolving structure, wherever those pieces may migrate to as changes occur.

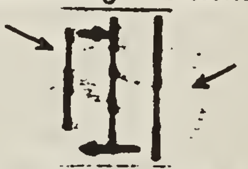
This simple facility-- the jump-link capability-- leads immediately to all sorts of new text forms: for scholarship, for teaching, for fiction, for hyper-poetry. This makes possible a certain free-form serendipitous browsing.





(continued from previous footnote)

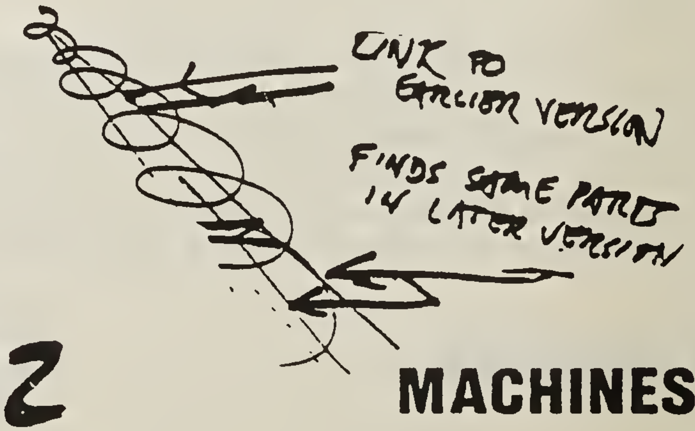
Thus the link stays where you put it through historical backtrack and in alternative versions-- if you choose to see it.


forward through versions;

backward through versions;
also "sideways" to alternative versions.

Essentially, the link seizes a point or span (or any other structure) in the Prismatic Document and holds to it. Links may be refractively followed from a point or span in one version to corresponding places in any other version. Thus a link to one version of a Prismatic Document is a link to all versions.

The effects, then, of links, alts and backtrack are in some sense multiplicative: together they give you a united facility of great power.

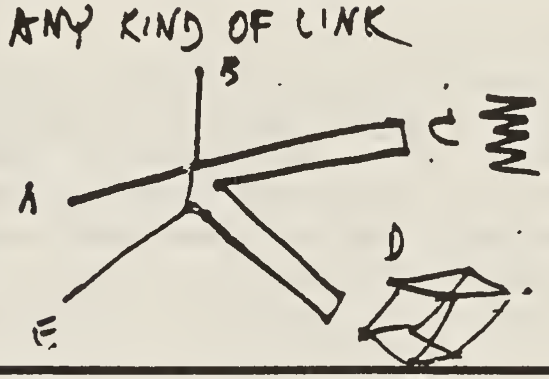


And any types of links may be created.

ANY TYPES OF LINK$

A proper system should allow any types of link whatever, and there are myriads of types.

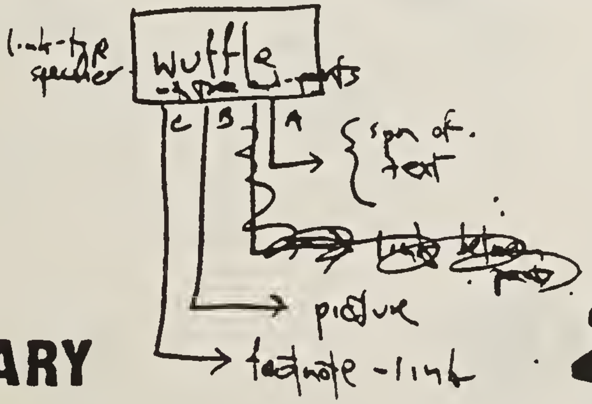
In principle we allow any types of link to be defined by the sophisticated user. These include point-to-point links, point-to-span, and span-to-span, having any separate names and functions desired. We also allow links with multiple endpoints.



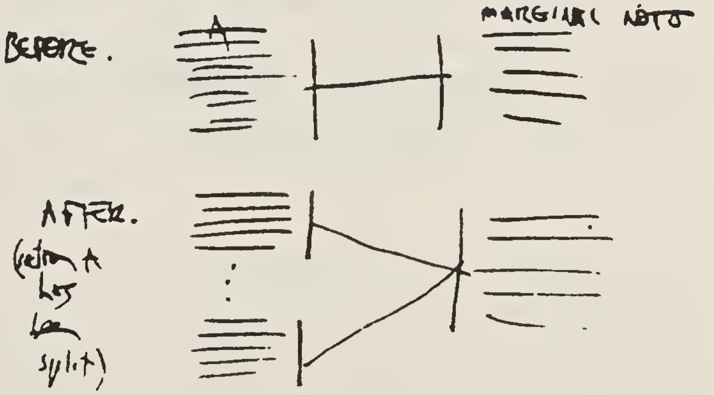
END-SETS

Links may attach to other links. Thus we have the concept of an end-set, the several types of object that a given link may attach to.

Consider, for example, an arbitrary type of link which we may call a "wuffle." A wuffle, let us say, connects a span of text, a picture, and a footnote. These are the endparts; together they constitute a wuffle's end-set.

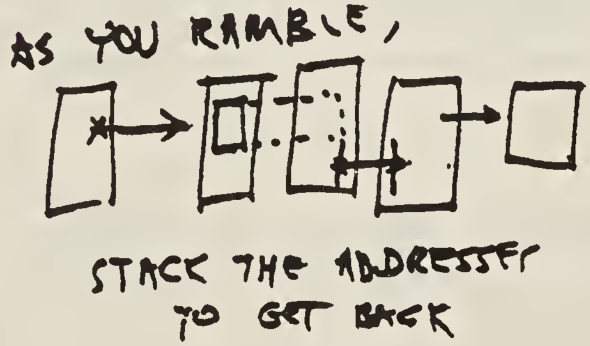


Directionality, if any, is given in the link-type definition. Note that end-parts may not hang together as they evolve (e.g. text sections):

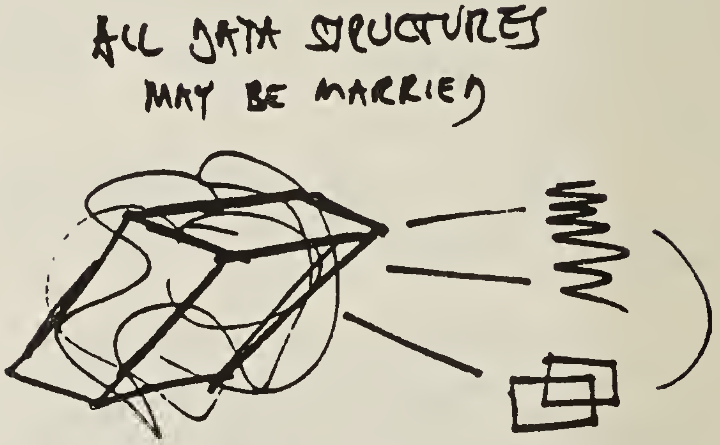


LINKS AND FRONT ENDS

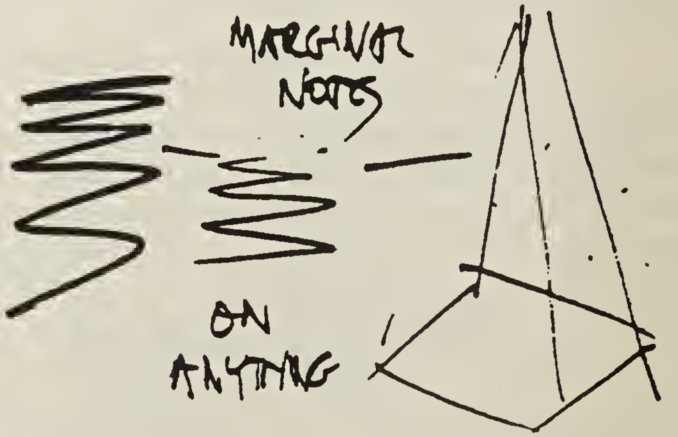
How to show links is a Front-End Function. So is the problem of keeping track of where you have been as you browsed; the front end must manage your stacks for you.



LINKING AMONGST ALL DATA TYPES



It is vital that a general system not have restrictions. For instance, why should you just have links on text? We believe you should be able to put footnotes and marginal notes on pictures, on music-- on any forms of data.



2.5 THE DOCUMENT CONVENTION

From these beginnings, it will be possible to create many levels of organization and overlay-- but first we will adopt by convention a fundamental unit.

So far we allow the storage of any sort of text (etc.) and any sort of links between points and passages of text. If anybody may put in and change anything, we stand to get simply a chaotic blur, a single muddy pool.

The solution is straightforward and traditional, and derives from literature as we have known it. We make sure that everything stored is divided precisely into separate documents.

An interesting choice has been made in the design of this structure.

A document consists of anything that someone wishes to store. A document is something designated by a person to *be* a document, containing text and/or links that he has created. We call a thing a "document" whether it contains text, or links, or both.

Thus the Gettysburg Address is a document; "Jabberwocky" is a document; and a set of links between them, were someone to create it, could be yet a separate document.

By this convention, then, everything in the system is a document and has an owner. No free-floating materials exist. What this convention really does is stress the singularity of each document, its external and internal borders. Thus, we focus on the integrity of the "document" as we've known it. Evolutionary continuity is unambiguous.

Every document has an owner. The rightful copyright holder, or someone who has permission from the copyright holder and pays for storage, is essentially the owner as far as the system is concerned. Only the owner has a right to change or withdraw a document.

We will call this unit a "document." It has an owner and (ordinarily) a name.

It normally consists of *contents*-- text, graphics, music, etc.-- and links to other documents.

(Although there are ways that others may conveniently have the use of changed *versions* that suit their needs, as we will see in a later section.)

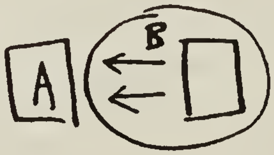
Ancient documents are owned by the system-- or preferably by some high-minded literary body that oversees their royalties (to be discussed later).

OWNERSHIP OF LINKS

Links may be created within or between documents. But each link resides in one place.

Links, just like text, are owned. Every link is part of a particular document and has an owner.

However, links in one document may attach to another document.

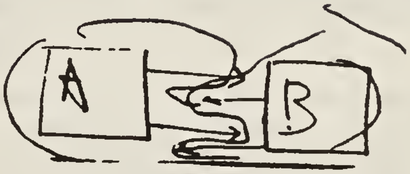


Every link, then, is part of a document.

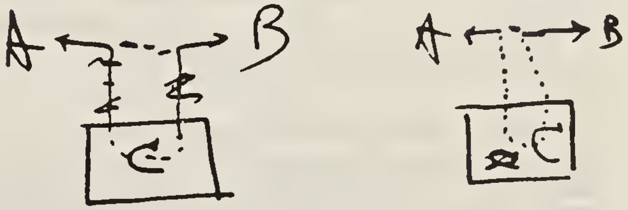
Putting it another way, a document consists of its contents and its out-links.

And that's all.

Ordinarily a document consits of its <u>contents</u> (including history and alternatives) and its <u>out-links</u>. These out-links are under control of its owner, whereas its in-links are not.*



Links between documents ~~text~~ may reside .. yet others



* More elaborate cases are possible, however. For instance, links between documents may reside in yet others.

2.6 COMPOUND DOCUMENTS

This ground rule allows us to have complex multi-level document structures-- criss-crossing superdocuments of many parts-- collected in new structural wholes.

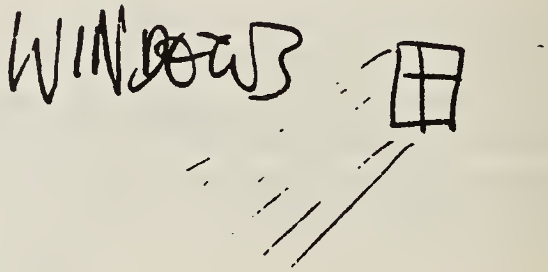
Once you have the package, the docuplex that allows linkage and backtrack, why not extend it?

Why not allow anyone to create links between documents, allowing you to jump straight from one to another?

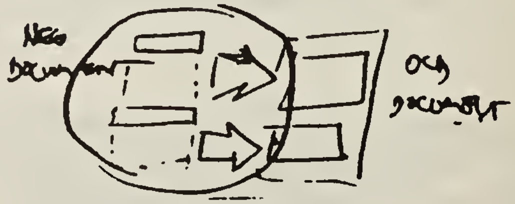
Given the exact document boundaries and ownership already mentioned, we can now create an orderly arrangement permitting far more complex documents to be stored. We also provide an arrangement allowing other individuals freely to make their own modifications on the stored documents. This we do by allowing so-called "compound documents."

The logic of these compound documents is simple and derives from the concept of document ownership. Every document has an owner. The integrity of this document is maintained; no one may change it but the owner.

But someone else may create a document which quotes it as much as desired. This mechanism we call the quote-window or quote-link. Through a "window" in the new document we see a portion of the old.



Thus a new document may consist of the quote-links and new material, if any.

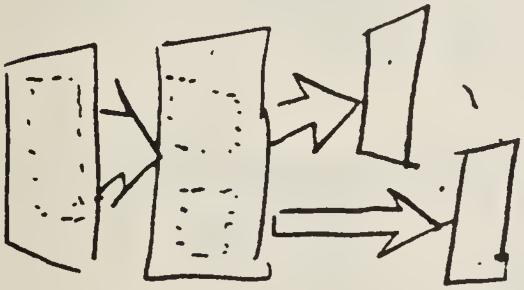


Through the same document conventions, the compound structures mentioned earlier maintain the same conventions of integrity and ownership.

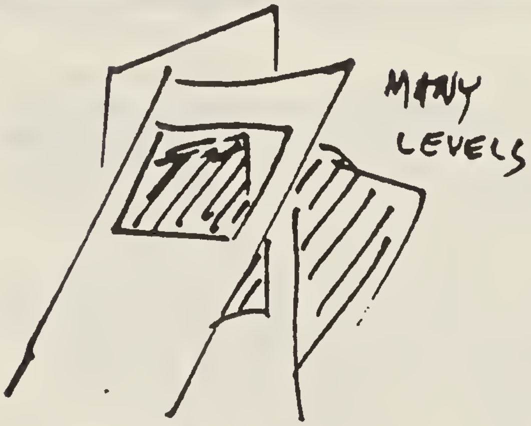
Each collection is likewise a document, and likewise has an owner.

The secondary document, too, has its own integrity, though the windowed materials are still part of the original document.

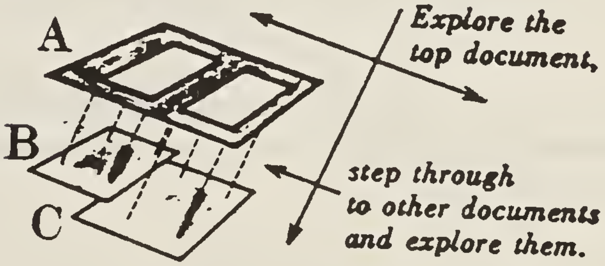
A document may have a window to another document, and that one to yet another, indefinitely. Thus A contains part of B, and so on. One document can be built upon another, and yet another document can be built upon that one, indefinitely: each having links to what was already in place.



Anything stored by one user on the system may be quoted-- adopted into a document-- by another person writing on the system (provided the second user has legitimate access). This freedom of windowing applies, of course, to all forms of data, including pictures, musical notation, etc.



Think of the present document as a sheet of glass. It may have writing painted on it by the present author; it may have windows to something else, in turn made of more layers of painted glass, with writing on each.


A
B
C
Explore the top document,
step through to other documents and explore them.

A reader may either explore the immediate document, or "step through the window" to explore the next document, or the one beyond. After exploring a further document, the reader may return to the one that showed him into it, or proceed on tangents that become available.

Document A can include Document B, even though Document B is owned by someone else.

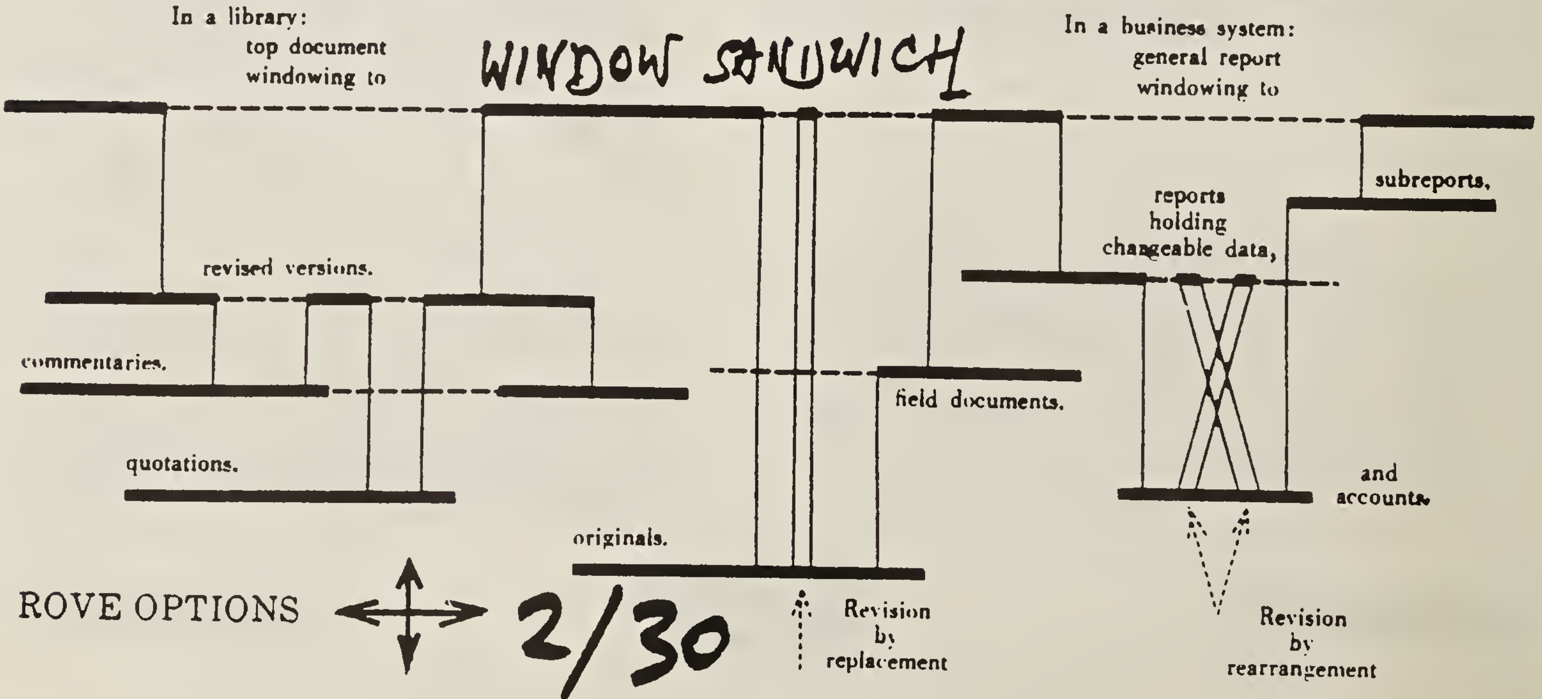
By this simple, sweeping mechanism, all manner of different requirements and specialized uses are reduced to a single structure. Each layer of windows may have, as it were, colored cellophane or opaquing on it. Only when you step through the window-- which you always may at any time-- do you reach the original. But stepping through the window means turning one glass page and going on in the next. Now you are in another work.

Example. The annual report of a corporation has a brief paragraph on every division of the company, with summary operating figures for the year. These paragraphs and figures are quoted from other documents which explain the matters more fully; the reader may easily step through to study them further.

Example. A children's story is illustrated with pictures. If the child wants to "reach through the window," each picture is found to be part of a larger picture, with another story attached.

The windows of a windowing document are themselves actually particular links between documents. No copy is made of the quoted material; rather, a quote-link symbol (or its essential equivalent) is placed in the stored symbol-train of the quoting document. This quotation does not affect the integrity or uniqueness of the original document, since no copy is made. Nor does it affect the ownership.

LAYERS OF WINDOWING TEXT. Each horizontal line is a document.

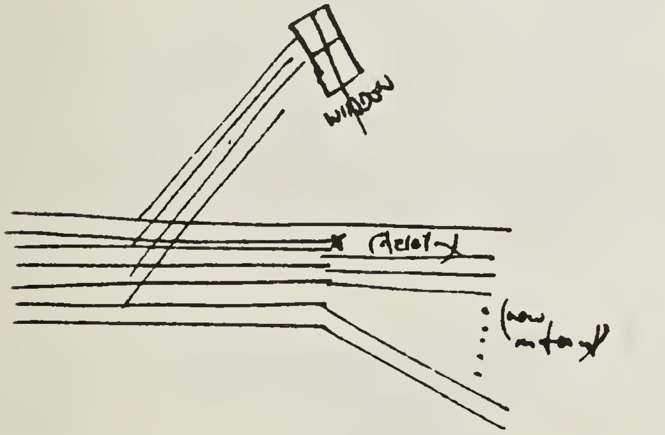


(Note that these methods of storage save a great deal of space, if the same material is used in numerous documents.)

The use of the special links dramatically simplifies a host of problems.

No copying operations are required among the documents throughout the system, and thus the problems of distributed update, so familiar throughout the computer world, are obviated. (But they do reappear on a later level.)

Since quoted material only has to reside in its place of origin, and not in the other documents that quote it, other documents that quote it may be automatically "updated" when its owner changes it.



Note also, however, that a window may be fixed to a document at a certain point in time, in which case revisions are seen by the user only when he asks, "What has this passage become?"

DERIVATIVE DOCUMENTS

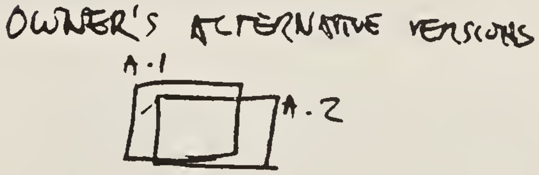
The integrity of each document is maintained by these separations: derivative documents are permanently defined in terms of the originals and the changes. (And stored on that basis.)

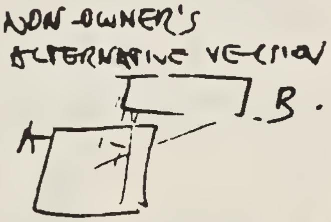
A document may consist merely of changes to another document. Thus the modified Gettysburg Address published in MAD by Doodles Weaver may be thought of as two documents: the original, and the changes.

ALTERNATIVE VERSIONS BY NON-OWNERS

A document owner may create alternative arrangements of the same material, all within the same document.*

Another user, however, is free to create his own alternative version of the document he does not own. This, then, becomes a windowing document using the same materials.

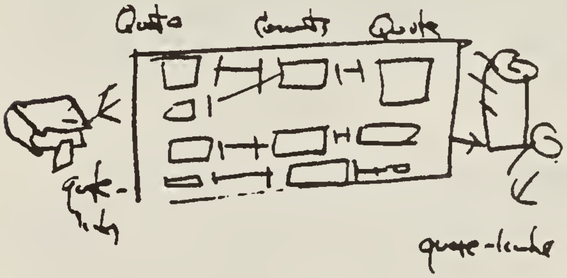




* The official naming-mechanism of the system has "document" and "version" fields. See "Tumbling through the Docuverse," a Chapter 4.

INTERCOMPARISON DOCUMENTS

A document that points out relations between other documents we may call an "intercomparison document." Such documents may be easily created, say, to point out relations between the Bible and the Dead Sea Scrolls.

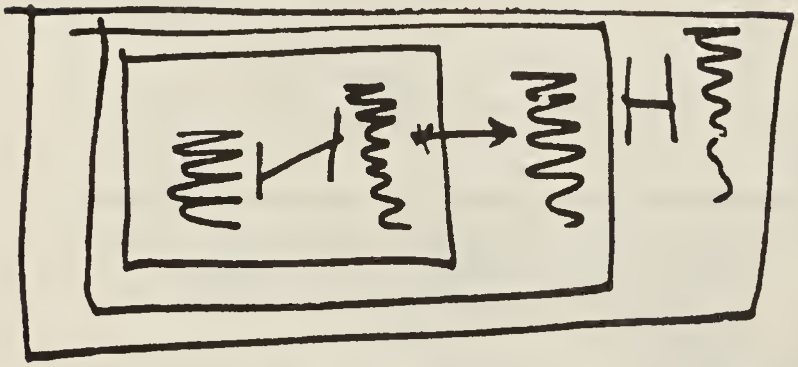


COMPOUNDING OF OTHER LINK TYPES

Any other link types (beside windows) may likewise go from one document to another, and interweave with quote-links.

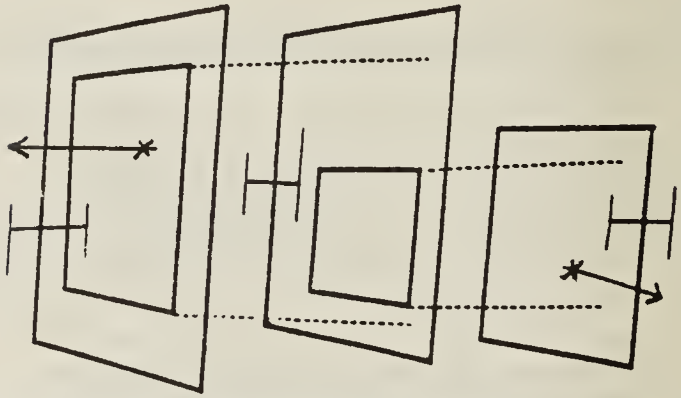
Note that links, like text and picturs, may be quoted.

These structures may of course nest. This makes possible compound documents to any remove, where one document links to another, and so on. One document, embracing another, takes it into itself.



... TO ANY REMOVE

And this creates a basis for all kinds of hypertext-- linked, parallel, windowing.

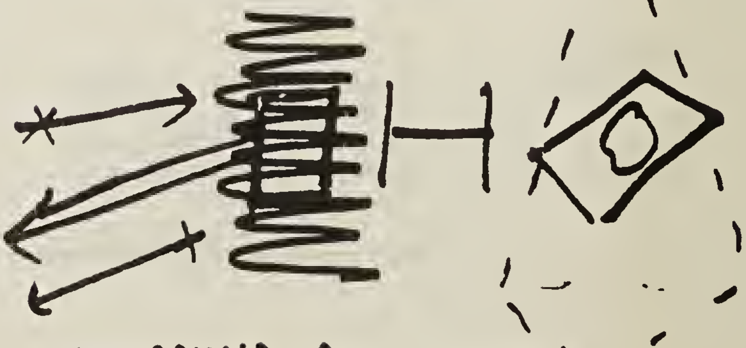


COMPOUND NESTED LINKS

AN INTERESTING WORLD

It will be noted that we have here defined an interesting and rich sort of world-- a world in which we are relieved of complications from conventional computer filing; yet we have greatly enhanced abilities to specify and express compound relations of every sort.

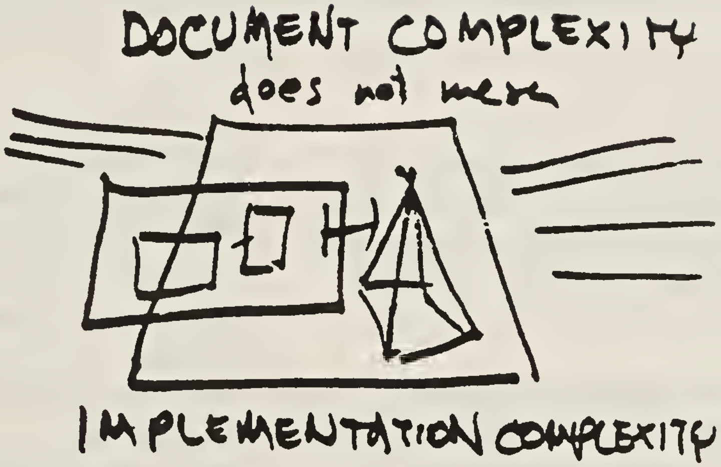
A WORLD OF INTERACTINGTWINGLED



COMPOUND DOCUMENTS

... BUT A SIMPLE ONE

This world nevertheless remains simple in design. The virtuality is simple in structure and repeats in layers.



27 ELECTRONIC PUBLISHING

MAKING THE LITERARY SYSTEM UNIVERSAL

Beyond its use as a private facility, we intend that this system be usable as a publication system. Thus a carefully designed system of publication, surprisingly like that of paper, has been worked out.

The orderliness and power of this approach are very suggestive. Given the hyperfile with links that we just expounded: Why can't we extend it into a full publishing system?

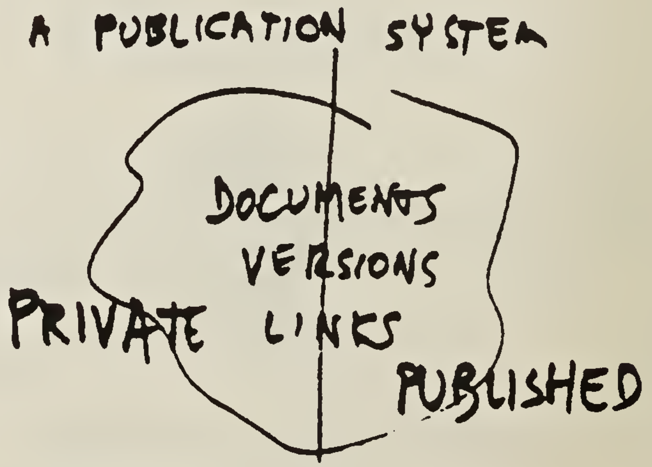
Suppose that the hyper-documents already stored could be reached and used by anyone. All that we need additionally is the ability to create links among them-- to make your own bookmarks and marginal notes, to quote from them by direct excision. And why not, indeed, allow users to publish assemblies and collections of compound documents building on the others?

Very well. Let us try to put together a publishing system-- that is, an overall arrangement where some of the documents stored in the "true" structure I have described may be made available publicly.

What does this mean exactly? Well, a publishing system, as we see it, should include provisions for privacy, copyright, royalty and accounting.

PUBLIC AND PRIVATE

The idea of "publication" in this system, as it is clarified below, will show what we mean by both private and public documents.



We can therefore have a system of electronic publishing that feeds to your computer screen exactly what you ask for, as soon as you ask for it; with royalties divided between the document owners in exact proportion to how much of their materials are transmitted or used.

A document may be private or published. A published document may include text, links, alternative versions and historical backtrack. But it need not.

Any user may store anything on the system. Unless specified otherwise it is a private document. A private document may be read and linked-to only by the owner and his associates. A published document is available to anyone, and may be read and linked-to by anyone.

INTERNAL COPYRIGHT CONVENTION

To bypass legal problems, we intend to establish a copyright convention internal to the network and agreed upon by all participants. To wit, if you publish a thing through the network, you have to agree to the same rules as everybody else-- which are intended to create a fair balance of incentives. More on this later, when the choices described here will be further discussed.

LINKING TO WHAT-HAVE-YOU

Any user may read, or otherwise employ, any published document on the system, or any private document to which he has legitimate access. He can make any kind of links to it from his own cocuments, private or not.

NO CONTROL OVER IN-LINKS

Accessibility and free linking make a two-sided coin. On the one hand, each user is free to link to anything privately or publicly. By the same token, each author of a published work is relinquishing the right to control links into that work. This relinquishment is part of the publishing contract.

"Private" materials are available only to their owners or designees; "published" materials are available to anyone, yielding a royalty to the owners.

THE ACT OF PUBLICATION

If the user chooses to publish a document, he may do so with relative ease, making it available to anyone throughout the network. It is then a published document.

Because publication is an important act, both for authors and readers, we make publication a solemn event, to be undertaken cautiously. Since publishing can be instantaneous and unrestricted, merely a "publish" button on the console could do it-- but the dangers of rash publication to an individual's reputation and career could be great. Some formalized technique would therefore be required for "committing to publish"-- probably a ceremony and signature; presumably signing a contract on something very like a credit-card triplicate slip. The author signs an "I hereby publish" form, after which not only is the document universally available, but he can't withdraw it.

NO WITHDRAWAL; SUPERSESSION; OTHERS' LINKS

Publishing is a solemn act, and it is in the common interest that a thing once published stay published, as in the world of paper. Consequently its author may not withdraw it except by lengthy due process.

However, the author may readily publish a superseding document, but the former version remains on the network. This is vital because of the links other users may have made to it-- which can reach through from the previous version to which they are attached into the newer version.

SUPERSESSION WITH NOTICE

It can easily be assured that fresh readers turning to an old version of a document may be directed to the new, unless they have reason to consult the old.

Private documents can link and window to public ones.

PRIVASHING

An author who wishes to render his work universally available, but wishes also to retain the right to withdraw it at any time, has a simple means for so doing. He simply designates his document as a private document with unrestricted distribution. Anyone may have access to it or use it, but the owner is free to withdraw it or change it irrecallably at any time.

No royalty is received for the use of privashed documents.

PRIVATE SALE OF INFORMATION

Those information purveyors not satisfied with our private rates are free to store what they like as private encoded documents and then sell access, or sell the secret code to make them readable, or whatever. This is a private transaction and does not involve the system.

ROYALTY FOR USE

In our planned service, a standard proportional fee is paid automatically by the user to the owner every time a fragment is summoned. Each owner must consent to the standard royalty-- say, a few cents per screen hour-- and each reader contributes those few cents automatically as he reads along. If the back-end cost of servicing a user is, say, $2 per screen-hour, then the royalty will be 5¢ to 10¢ for that hour. This is deducted automatically from the back-end fees.

Since there is no controlling what happens at the user end, this royalty should be largely based on transmission time. An hour, five minutes, or one second of a thing, each contribute proportionally to the copyright holders' accounts. (The question of whether to allow different rates of royalty will be discussed later.)

The royalty goes to the owner. We say "owner" here to avoid having to distinguish between authors and publishers; but this is a matter of private arrangements that is of no concern at the system level.

This is a radical and daring idea: a new form of reading and writing, in a way just like the old, with quotations and marginalia and citations. Yet it will also be socially self-constructing into a vast new traversible framework, a new literature.

THE REAL POWER:
PUBLISHING COMPOUND DOCUMENTS

The windowing approaches already mentioned automatically furnish a general solution to the "copyright problem" with regard to quotation and citation, simply by this means: authors who are windowed automatically get royalties as well. If the windowing document is electronically published, royalties are paid to the owner of each document seen in proportion to its use.

Since the copyright holder gets an automatic royalty, anything may be quoted without further permission. That is, permission has already been granted: for part of the publication contract is the provision, "I agree that anyone may link and window to my document." Publication through such a net requires your permission for your work to be quoted ad lib. You publish something, anyone can use it, you always get a royalty automatically. Fair.

PUBLISHING MODIFIED VERSIONS

You can create new published documents out of old ones indefinitely, making whatever changes seem appropriate -- without damaging the originals.

This means a whole new pluralistic publishing form. If anything which is already published can be included in anything newly published, any new viewpoint can be fairly presented. (Especially if the reader can always say, "Show me what this was originally.")

If a modified document is read, the original owner and the modifier split the royalty in proportion to who wrote what, as determined automatically. (For royalties on links, see "Tuning," Chapter 2.)

For example, my great-grandfather, Edmund Gale Jewett, believed that one word in Hamlet was incorrect. (It should have been siege of troubles, not "sea" of troubles, in the well-known soliloquy.)

Very well: if Hamlet is already on the system, then E.G. Jewett could publish his own Hamlet very easily: a quote-link to the whole original, except for "sea," which is changed to "siege."

Given that anything on such a network may be available instantly, such an arrangement promises an extraordinary new level of capability. For not only may simple documents be accessed at once, but compounded and windowing documents may be overlaid on anything -- promising a new degree of understandability through what is added later.

Now, the obvious rules of the road should be as follows:

1. Shakespeare's _Hamlet_ is of course unchanged and available instantly.

2. Jewett's modified version of _Hamlet_, composed almost entirely of the original, is also available instantly. (Jewett may give it any title he wants.)

3. Shakespeare-- or presumably the Authors' Fund-- gets the royalties for whatever of Shakespeare's _Hamlet_ is summoned by readers.

4. When people read Jewett's _Hamlet_, the Authors' Fund still gets the royalty on Shakespeare's behalf almost all the time. But Jewett gets a minute proportional royalty for the change he has made, whenever a reader encounters that part.

5. Anyone reading Jewett's _Hamlet_ can say, "Show me the original of this next to it," or just "Take me to the original."

6. Anyone reading Shakespeare's _Hamlet_ can say: "What documents have links to this?" or "Are there any alternative versions?" and get a list that includes Jewett's version.*

SECURITY RAMIFICATIONS OF WINDOWING

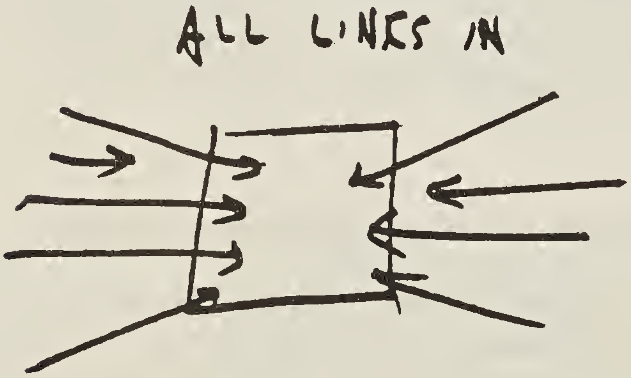
These ideas simplify the creation of rationally-ordered document dissemination systems with clear-cut security. A document can only window documents with the same, or lesser, security level. For instance, a private document may window a published document, but not vice versa.

* Note also the modest cost should Jewett publish this; the storage cost for a few hundred bytes (ID, pointers and changes).

It is our unusual hope and vision that this, with its simplicity of approach and efficiency of implementation, may become the standard publishing medium of the future.

SHOWING AND SIEVING IN-LINKS

The reader should be able to ask, for a given document or place in the document, "What connects here from other documents?"-- and be shown all these outside connections without appreciable delay.*



But there may be too many. Indeed, for "Alice in Wonderland" or the U.S. Constitution, the number could be in the squillions.

Thus it becomes necessary to apply some kind of filter, saying, "What links come in from Spain? From last week? From last year in Marienbad?"-- and see the number of such links at once, followed by the linked documents themselves if desired.

This must all be fast enough to please the impatient on-line user. And we believe it can be done. This filtering by different attributes we call "sieving"; and it can only be set up for a comparatively small number of traits-- say, location and author and time.**

* Technically knowledgeable readers may note that this is the hardest feature. This is the stopper. But we believe it can be done.

** Of course, any amount of additional sieving can be put in at the front end.

ALL END-USE IS LEGITIMATE

The user may employ any terminal, graphical or printing. Viewing-methods and manipulations are up to the terminal designer. No restraint is contemplated as to what use may be made of the materials found on the system, since no restraint is possible.

There is no way whatever to ascertain or control what happens at the users' terminals. Therefore _perforce_ all use whatever is legitimate, and anyone who plans to be vulnerable to "misuse," whatever he thinks that may be, had better keep his stuff off the system.

PRINTOUT AT USER TERMINAL

Users are thus free to make printed copies for their own use. Note that if a user prints out a document, he has _paid_ a royalty on its transmission. It is _paid up_, just as a paperback bought at the drugstore has had its royalty paid up. And if a copyright holder cannot be satisfied with this arrangement-- even knowing no other is possible-- he had better withhold his stuff from this system.

Note also that he who makes a paper copy is losing all dynamic link connections, and is left with the inert, noninteractive copy. And while some of you may have trouble believing it, that will be a considerable deprivation in the world we are talking about.

WHAT'S A PUBLISHER?

Traditionally, an author is someone who creates a work, whether or not on his own initiative. A publisher is a businessman or business firm that takes the business initiatives, deciding to publish the work, fronting the money to print, warehouse, and advertise. He, or it, also assumes certain legal risks from which he usually guards the author. The publisher may also encourage and cajole the author, buck him up, advance him money for work unfinished, and finally erect a plaque to his memory on the Hotel Chelsea.

Ownership and copyright are split between author and publisher according to their own negotiated private arrangement.

If compound united hypertext is the printing press of the future, the publisher of the future can do all these things in exactly the same way. Except now there is no "printing and warehousing," but a certain required minimum disk rental. Thus a "publisher" is someone who pays for the rapid accessibility of materials and benefits from their use along with the author.

ON-LINE BOOKS AND MAGAZINES

Just the fact that things are on an electronic system does not threaten their integrity; indeed, every unit fully retains its integrity in the present system.

A book is still a comparatively large unit of writing or anthology, written and published by specific peopel on specific dates.

A magazine or journal is still a collection of shorter pieces (perhaps windowing from elsewhere) which is regularly edited by the same person or people, and regularly published at a specific time.

Magazines and journals will have great importance in such a publishing system (as they do in the paper world) because they will furnish stabilized views of the world, offering a predictable kind of material, and bringing in, evaluating, and now windowing, ideas from all over.

DIRECTORIES AND CATEGORIES

Two system directories, maintained by the system itself, are anticipated: author and title, no more.

Other directories would essentially involve categorization, like the Dewey Decimal and Library of Congress catalog systems, or the Yellow Pages of the phone book.

There is nothing wrong with categorization. It is, however, by its nature transient: category systems have a half-life, and categorizations begin to look fairly stupid after a few years. (Indeed, simple categorizations of computer articles in computer bibliographies of ten years ago have already begun to look stupid.) The army designation of "Pong Balls, Ping" has a certain universal character to it.

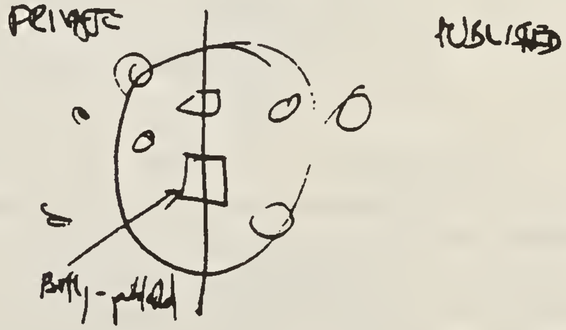
All category-systems make some sense, few stay good for long. (However, the Yellow Pages categories are an interesting exception, being dreadful to begin with, and, though supposedly updated from time to time, do not seem to improve. Try to find from them the nearest place to make paper copies.)

What is the solution for our system? Keep categorizing directories <u>out of the system level</u>. This is user business; let them handle it and collect royalties.

Provision will exist for anyone to publish his own document lists, categorized in any way he imagines, and have users bounce through them in search of whatever they think they may find.

PARTIAL PUBLICATION

A whole document need not be published. That is, someone may publish what to him is only <u>part</u> of a document, keeping the rest private. Very well: now his <u>private</u> document is windowing into the materials he has published as part of the "real" larger document that still appears on his screen.

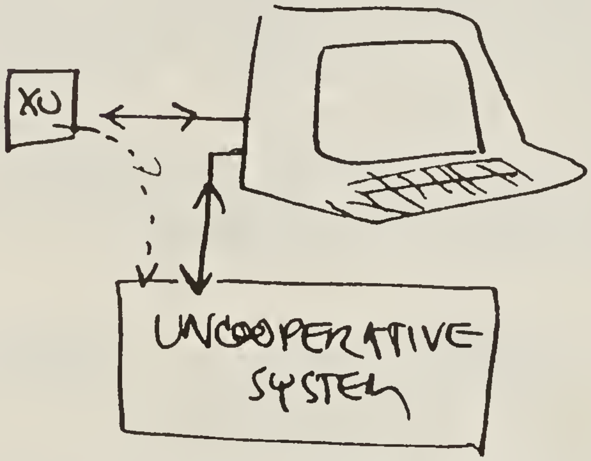


Historical backtrack alts, etc., may also be held privately in a common pool with documents which have entered the "published" world.

USE WITH OTHER DATA STORES

There are many purveyors of lesser services who say, in effect, "Ha ha, we've got the copyrighted material, you can go hang."

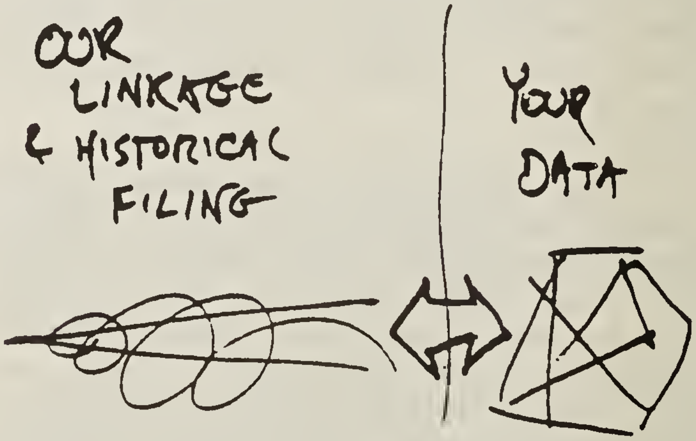
Maybe yes, maybe no. Suppose that Company Y has some key legal document on line-- say, the Napoleonic Code-- and you need to make marginal notes; they offer no such facility.



Well then! Connect to ours and theirs at the same time. Write your marginal notes on our system, with the linking information; then your front end can call up the Controlled Document and show it with the notes you've stored on our system.

In the general case, then, we can marry our data structures and linking facility even to the dog-in-the-manger on-line material whose purveyours do not wish to cooperate.

Or suppose you have private material you do not wish to expose through communication lines, even in encrypted form. Nevertheless, the system we are discussing can help you with detailed linkages, backtracks, etc., even though it runs on a public system: for you may use its indexing facilities to control your data, sight unseen. Your data stays home, blind indexing is stored on the net under your control-- with the contents it controls wholly unknowable to any parties but you.



VIDEODISC CONNECTIONS

There has been a great deal of whoop-te-do recently about videodiscs, the storage devices that hold one or more hours of TV on a platter. Several of these are now available and incompatible. Some of them offer freeze-frame and random frame addressability. Very well: they are a fast image playout that can be hooked up to our indexing for complex purposes.

(The widely-touted notion that videodiscs will be useful for text libraries seems a little silly, since they make it possible to access only what you actually have right there, while a hypertext network could allow immediate access to everything on it; a vast difference.)

((Another use of the term "videodisc," causing total confusion, is its use to refer to certain high-density write-once digital disks under development by Phillips. We are often asked whether these "videodiscs" will be useful for our system, and the answer is yes, but they aren't videodiscs.))

2.8 DISTRIBUTION AND NETWORKING

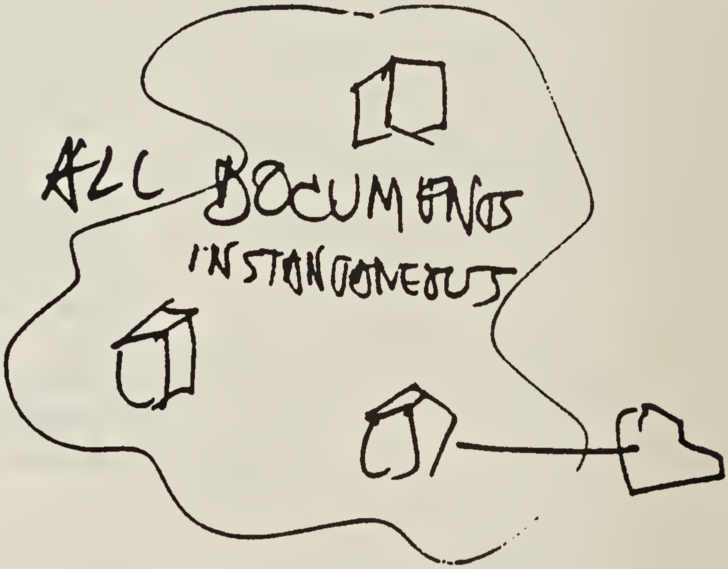
It might be possible to do all this out of one feeder machine, but there are disadvantages.

In the previous sections of this chapter we have discussed virtuality, in particular, the conceptual structure of the proposed system. So far, the service might be provided from a single computer, or "centralized data bank." We have more or less assumed a single-processor version of this system, one which easily treats all documents and their versions as an interconnected whole because they are stored in the same place.

However, there are fairly definite limits on what one machine can hold and the number of users it can provide services to. For the services described here to be seriously expanded to large numbers, it will be necessary to "network" the service through multiple computers distributed throughout the nation, and/or the world.

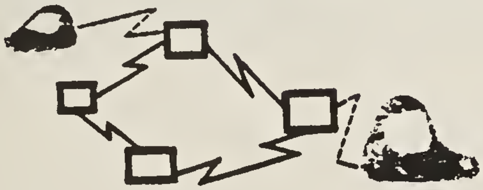
The system should be able to grow without size limit, containing in the body of available writings whatever anyone has stored from any place on the network.

So we turn to the idea of storing the materials on a network of computers. But we do not want the virtuality to change. A user should get anything he asks for an instant after the request, even if it comes from far away-- however widely scattered its parts may be in their storage and ownership, even if some parts of it come from one faraway place, other parts from other faraway places.



All of storage near and far becomes a united whole-- what is now called a "distributed data base." Actual locations are essentially invisible to the user; or, in that traditional phrase, "You don't care where it's stored." The documents and their links unite into what is essentially a swirling complex of equi-accessible unity, a single great universal text and data grid, or, as we call it, a "docuverse."

In principle it is possible to extend this system of storage and publication to a whole network of feeder computers.

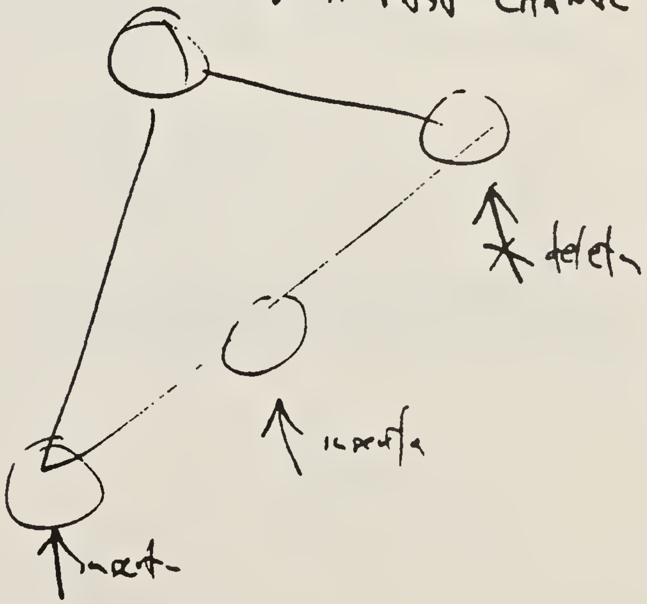


A user at any location may store what he or she wishes; links may be created by anyone, from anywhere, to bring a document (or part of one) to the inquiring user. Given today's network technologies, this is not really difficult; this part of the system-- the immediate delivery of anything from within a large arbitrary network of computers-- is not far-fetched. From a nuts-and-bolts point of view the material is more efficiently dispersed among holding stations united by a communication network. This is essentially state of the art.

From the software point of view -- needing to unite the documents into a single, instantly-available docuverse-- a number of challenges exist.

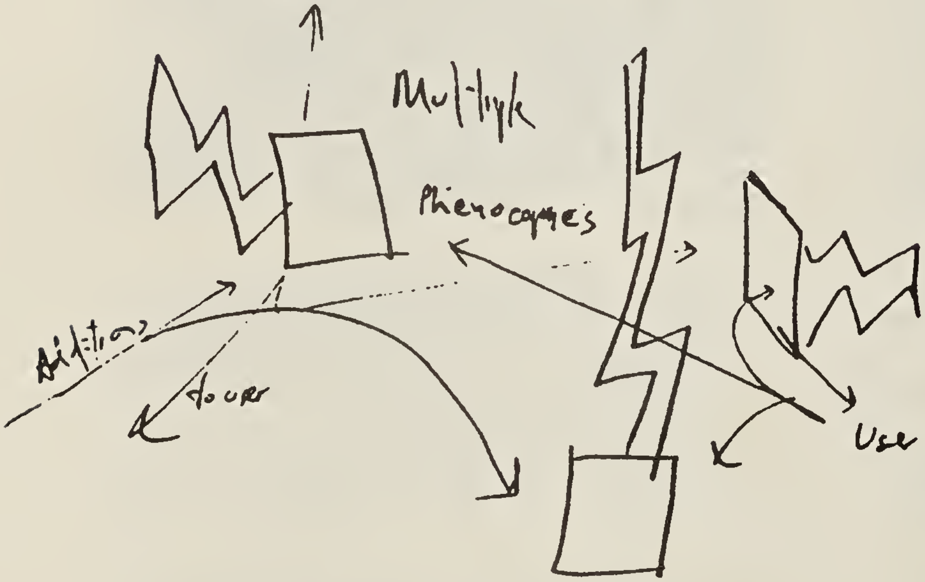
One is that multiple copies of each document must be distributed about the network for safety-- in a shifting distribution that keeps up with demand and other needs. Another is that these copies-- even with their historical-trace backpacks-- must be updated in place. And a change must somehow be known throughout the network the instant it happens, with new things at once assimilated to the great corpus.

MULTINODE DISTRIBUTION
WITH FAST CHANGE



The stored literary contents of all the computers on the network may be continually united into a single, accessible whole.

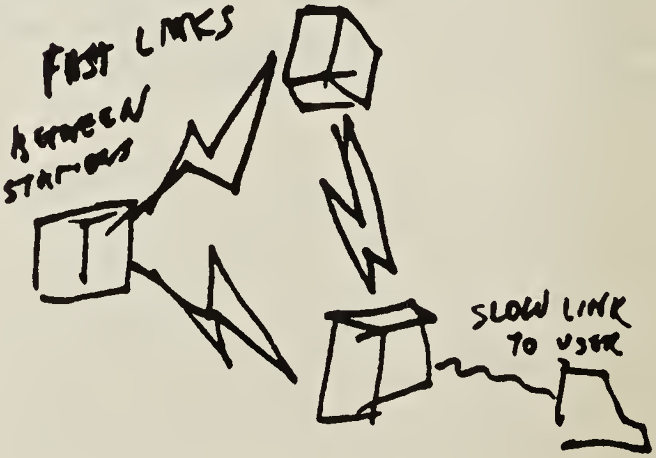
Moreover, while copies of all information cannot be stored in each location, enough of a trace or string must be in each place to pull in whatever is needed from wherever it is-- a "ripcord" to unleash any selected document.



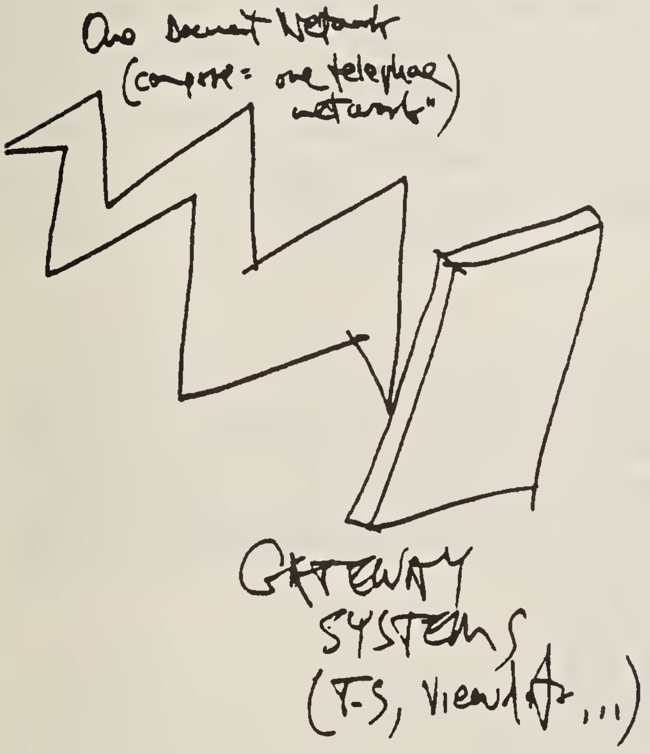
(Schemes for all these needs exist, having been worked out within our proprietary data-structure framework.)

NETWORK CONNECTIONS

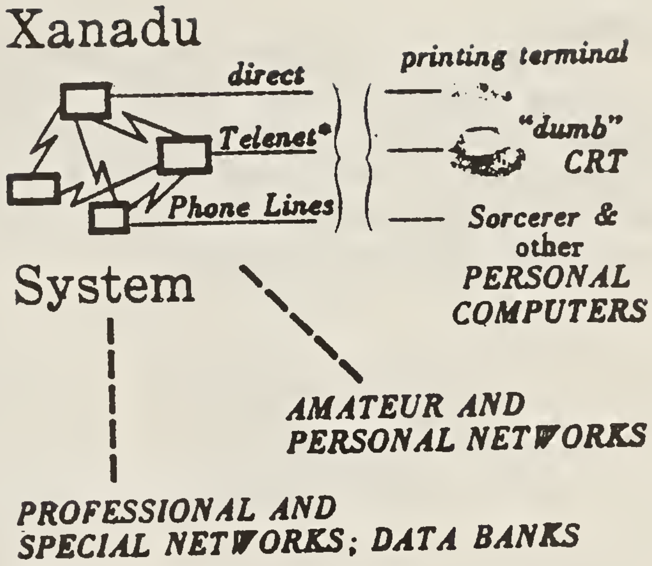
Essentially the network will have two connection speeds: the fast lines that unite the stations, and the slow links to users.



For connection to users, the contemplated network should of course hook up by whatever channels are most convenient for them: directly (at Stations), and via communication links such as Telenet* and telephone. It should also tie to other digital networks, either in the amateur market (such as The Source and PCNet), or the professional world (such as ARPAnet).



It is intended that the contemplated service will hook up to all ethical vendors who wish to offer gateway service to our docuverse.

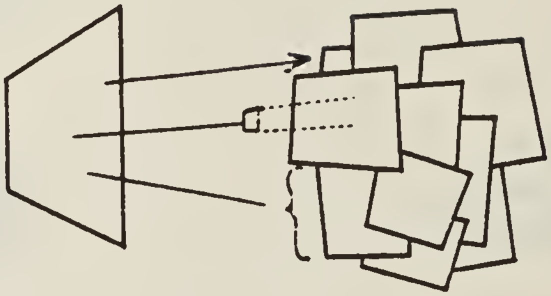

Xanadu
System
direct
Telenet*
Phone Lines
printing terminal
"dumb" CRT
Sorcerer & other PERSONAL COMPUTERS
AMATEUR AND PERSONAL NETWORKS
PROFESSIONAL AND SPECIAL NETWORKS; DATA BANKS

For the connections between stations, the mechanics of computer networking are fairly straightforward, and we need not go into them here. The so-called "packet" approach (now being standardized under the name X.25) allows direct commercial hookup via Telenet. More high-budget and high-flying approaches can use direct satellite links between stations, which are available and feasible now.

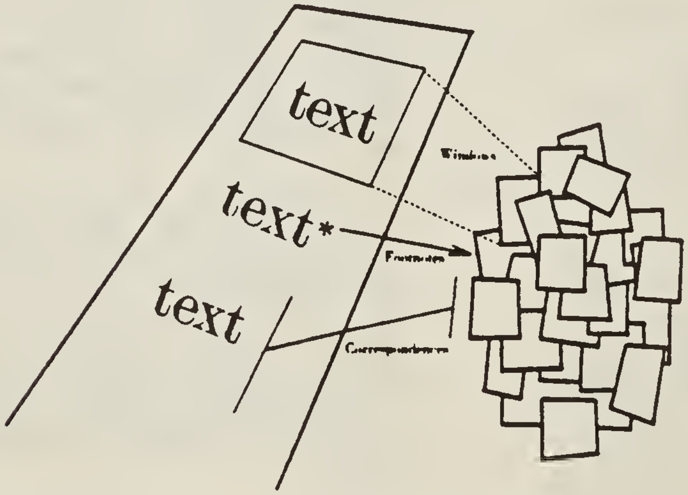
"How big will the total storage be?" people ask. The answer is, as big as people will pay for. Everything stored has to have money behind it. The system will grow as long as paying demand increases-- which should be for a considerable period. No matter how big it grows, you will be able to get anything in it very quickly-- as long as disks are added to the system.

Example:

A scholar writes a new interpretation of ancient Greek society, with numerous quotations from the writings of those times. Each quotation is a window, allowing the reader to step through and read the original.



ANY SUBSET, SUBSTRUCTURE OR GROUPING



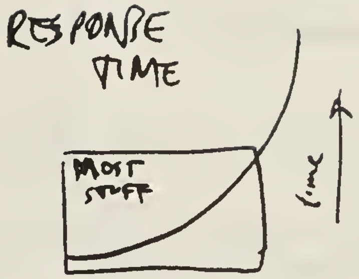
WINDOWING INTO THE PRIOR LITERATURE—previous public contents of entire network

THE GREAT HOPE AND CONJECTURE OF INSTANT UNIFICATION

Perhaps the most important thing, and certainly the hardest part to believe, is that everything on such a network can come immediately, even as it grows to tremendous size.

First, let's be clear what we mean by "immediately." We mean very quickly, even though there will be variations. If you telephone San Francisco from Chicago, you get through "immediately"-- that is, within perhaps three to fifteen seconds, with an off chance of having to wait thirty. We're talking about figures like that. *

In any case, for comparatively local service (on-planet or nearby), we may look forward to "instant" retrieval of whatever is asked for. This means essentially that all documents become a single instantaneous whole.



This in turn means that compound documents of any kind become effectively as accessible as simple ones. And we may read and write accordingly.

* Now, as human habitation grows beyound the planet and speed-of-light considerations become significant, obviously performance will degrade. For off-planet users (or earthbound users calling an off-planet station), the usual performance figure must be added to the transmission time. Obviously, too, if interstellar travel is ever achieved, transmission delays will degrade response time to months and years. (However, some thought has been given to this problem in the overall design: see "Tumbling through the Docuverse," Chapter 4.)

2.9 VITAL ISSUES

Thus we have the framework of a complete, radically different way of handling information.

USER PRIVACY: A VITAL ISSUE

The network will not, may not monitor who reads what or who writes what in private documents. This is vital. It is not easy to guarantee and impossible to make fully automatic. This important problem will be taken up later, in Chapters Three.

A PRINTING PRESS

We consider that this system may best be considered as a "printing press" of the future.

FREEDOM OF THE PRESS

If this system is a printing press, we can brook no greater restriction on its functions than on conventional printing. Freedom of the press has been challenged by tyrants and scoundrels since Gutenberg. It will happen again, and worse, on this new playing-field. We must be ready.

LEGAL GOOD BEHAVIOR

Plainly, the system must live within the law. However, what the law _is_ may often not be clear. Grey areas (for the USA) involve pornography, libel, and "national security" (often meaning matters embarrassing to a political administration).

There is no thinking out all these eventualities. But this is a libertarian system: restrict it, and all will lose.

Numerous issues of personal freedom are conspicuously present.

JOHN DOE PUBLICATION

Normally publication will be by stable individuals and companies with known residence or place of business. Still, there is no reason that anonymous publication by walk-in and transient users of the system should not be allowed.

PEREMPTORY CHALLENGE OF JOHN DOE PUBLICATION

However, John Doe publication is more sensitive to challenge, since the John Does do not hang around to defend their acts.

Hence peremptory challenges of John Doe publications must carry weight.

Consider libelous or uncomplimentary John Doe graffiti defaming specific individuals. If John Doe is not available, the affected individual should be able to effect removal of the materials by peremptory challenge.

CHALLENGE OF DEFENDED MATERIALS

However, where materials are published by stable and accountable individuals or firms, peremptory challenge no longer holds water, and removal must be by negotiation or by court order.

What we call "tuning" the system is the development of simple, fair and well-balanced arrangements and pricing that will balance users' incentives for the flexible and reasonable use of the system. (See "Tuning," Chapter 4.)

COPYRIGHT VIOLATIONS

Once materials are outside the system and the user's terminal, normal copyright law applies. Thus making magnetic or paper copies of documents, and giving them away or selling them, is tortious and illegal, as it would be without our system. Enforcement, of course, is the victim's problem.

Since the use of any material on the system by windowing is defined as fair use, people are much freer to do what they like with whatever is available.

One form of copyright violation is quite esoteric. That it is frequently mentioned suggests that people are reaching rather far for objections.

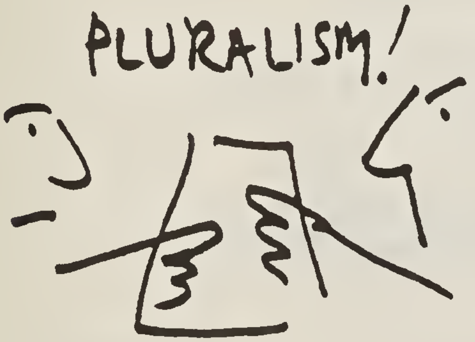
It is this: what if someone *makes a copy* of materials published by someone else on the system, then *re-enters them as his own in order to obtain royalties*?

As long as it went undetected, this scheme would work. However, the violator is exposing himself, if detected to a prima facie case of copyright violation.

PLURALISM-- AND NEW UNDERSTANDING

What is in such a publishing network may be revised by anyone, reinterpreted, redesigned. Anyone can publish a new version of Thomas Aquinas, Ayn Rand, Einstein, or whoever else's writings are on the system, attempting to reach the true and correct formulations that always seem to elude the person ahead of you. And no harm is done, no credit lost, to the originals.

The same applies to explanations. Most scientists and philosophers are not the ones to clarify their own work. The writings of a Niebuhr or a Talcott Parsons need to be considerably clarified by other commentators before most people can understand them.



Anyone may revise anything
— harmlessly

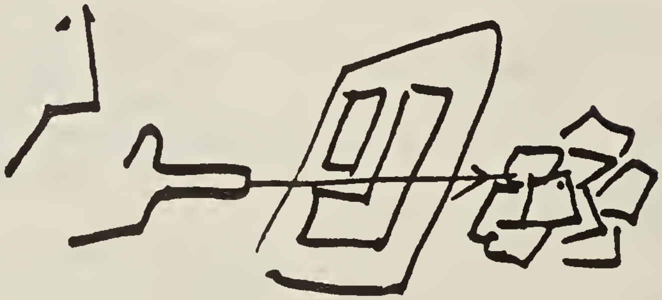
Very well. With the capacity for any number of compound windowing documents, good explainers-- the Asimovs of tomorrow-- can take what is already there, and add the many clarifications that will bring understanding.

Is this chaos? Not at all. Because at any one time, you are within one specific document, the work of a specific author. If this work is windowing to other documents, nevertheless you are not "in" the others, but viewing them through the present author's filter.

ETERNAL REVISION

There is no Final Word. There is always a new view, a new idea, a reinterpretation. Windowing hypertext offers the possibility that all writings (never mind the word "knowledge") may be forever revised and reinterpreted by new scholars, summarizers, popularizers, anthologizers.

ETERNAL REVISION



CHAPTERS THREE

SUMMARY OF THE XANADU™ HYPERTEXT SYSTEM

THE SILVERSTANDS™

BUSINESS PLAN

TOWARD A SUBCULTURE OF INTELLECT

FREEDOM IN OUR TIME AND BEYOND

CIVILIZATION AND ITS DISK-CONTENTS

SUMMARY OF THE XANADU™ HYPERTEXT SYSTEM

SHORTEST DESCRIPTION

It is a fast linking repository with windows and criss-crossing superdocuments.

MEDIUM-LENGTH DESCRIPTION

The Xanadu^{tm} Hypertext System is a fast linking electronic repository for the storage and publication of text, graphics and other digital information; permitting promiscuous linkage and windowing among all materials; with special features for alternative versions, historical backtrack and arbitrary collaging; with royalties for copyright holders and capable of indefinite growth.

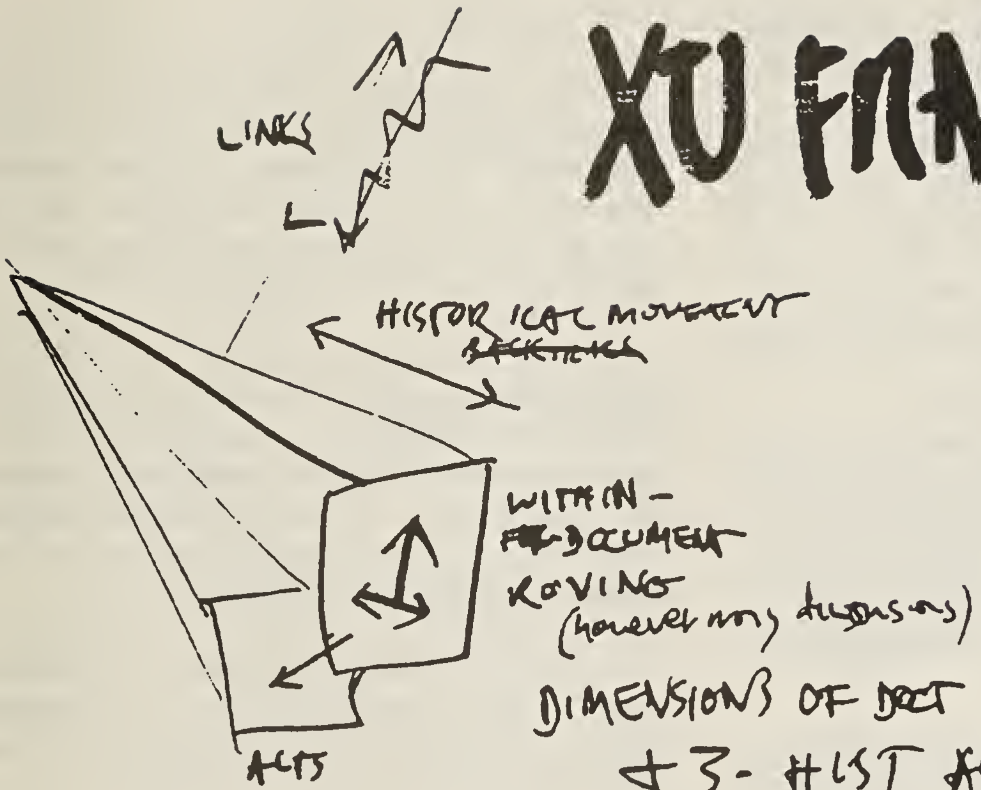
EXTENDED DESCRIPTION

The Xanadu^{tm} Hypertext System will be an unusual and probably unique repository in which all forms of material-- text, pictures, musical notations, even photographs and recordings-- may be digitally stored, at prices comparable to storage on other computer systems, and accessible from any port at any time.

Basically it is a system for the rapid delivery of linked and windowing documents and the assimilation and storage of changes. It is based on new technicalities which are of no concern to the user, and materials are stored in locations unknown to him. There are simple categories of publication (private and public) and a flat cost of usage.

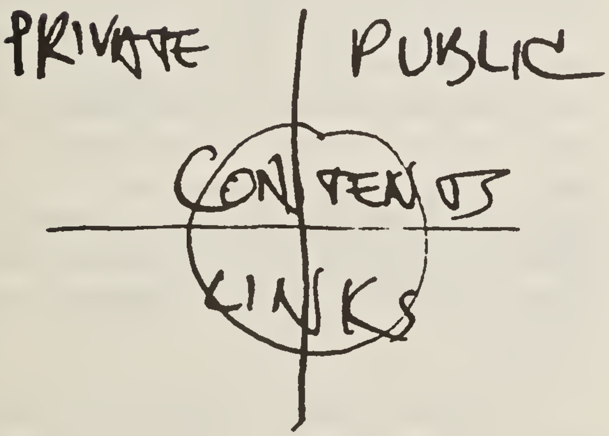
Its unique facilities of backtrack, linkage and windowing will allow the creation of new forms of multi-level, explorable collections and collages of material-- without losing the well-defined authorship and ownership of all parts.

XU FRAMEWK



DIMENSIONS OF DOCT

+3: HIST, ALTS, LINKS



There are four kinds of movement in a document: within the document's own topology (forward and back for simple text), plus historical forward and back, corresponding part in alternative versions, follow link.

Bit-map graphics are stored in such a way as to allow scrolling and zoom.

The objective is to build a powerful and fair system of storage and publication that will enhance our lives, gives us a start toward safeguarding our freedoms and heritage, and conceivably stand-- and expand-- for centuries.

Anyone may publish collaged and windowing documents having finely-divided ownership. It is believed that this will make possible a whole new universe of knowledge and understanding.

The actual code of the system is a medium-sized program in the C language, currently running under the Unix Operating System but expected to migrate to standalone. This software will not be made available in the currently-planned future, except in object form at a high monthly price, for Unix machines; VAX under Berkeley Unix preferred.

It's exactly one system, a back-end storage feeder with huge capacity and arbitrary topology. It's customizable at the front end only. It is not adapted for binary-riffle data bases of the card-catalog type.

The intended public operation of the system will be in a chain of suburban or roadside stations, called Silverstands^{tm}. New users will learn the operation of the system at such stands, and local users may dial into their nearest Silverstand.

Silverstand personnel ("Conductors") will include both local people and an itinerant corps of circulating smarties.

While any sort of terminal may be connected to the system, best operation requires a full computer in the user's terminal, programmed to handle display functions, interchange protocol, stacking and other work. Users may connect their home or office computers of any kind to the system, whether by dialup, GTE Telenet, leased line, twisted pair, or nearby wink-laser.

Xanadu service will be differentiated only with respect to speed of terminal (300 baud the minimum). No users will be restricted as to accessibility of materials. Anticipated initial price for 300-baud service is $20/hour prime time, $5/hour night, not counting communications or front-end costs, but including all feed and character input.

All users will have access to all public documents instantaneously.

Later prices in the Silverstands will aim lower, to prices like $5/hour (not counting terminal rental; a user may bring his own).

Disk storage prices will begin n the order of $200 per megabyte-month and drift downward as disk prices go down.

A variety of other services will be provided, including printout.

Publishing requires an up-front cost of one year's disk rental. A secondary publisher using windowed material need only pay the cost of pointer storage.

Materials no longer paying disk storage rates will be swung into tape storage for archival purposes, on some self-supporting generalized economic basis. Thus for slight extra cost and delay, earlier materials will be rapidly available.

Private docuemtns are available only to the owner and the owner's designees. Published documents are available to anyone, and yield a royalty to the owner; they may be updated at will, but the earlier contents remain available. They may not be withdrawn from publication without six months' notice except by court order. "Privashed" documents are available to anyone, and may be changed at will, but yield no royalty.

The system's contents will be supplied by customers only. There will be no participation by the Xanadu enterprise in the publishing process itself; neither contents nor indexing will be provided by the system, these being rightful endeavors of the customers.

The system will exert no supervision or censorship on stored or published materials, and court orders will be required for the removal of any material held in a stable account.

There are no special rules governing particular kinds of document (such as the "teleconference"). In all cases of functions which "require special rules" in other systems, we have found them to map clearly into our document structure or be movable into front-end functions.

In later phases it is contemplated that the full mechanism-- the actual program-- may be made available for the expansion of the standardized network.

THE SILVERSTANDS™

The Xanadu stations, or Stands, will be the local outposts of the network. Your Xanadu stand is where you'll start your account (even if you plan to be an at-home or office user), and learn to use the system. A cheery young person in futuristic garb will sit you down at a screen, and show you through an area of material of interest to you-- text or pictures. Then, at the moment of Xanadu Shock, when you get it, when you cry "Holy ----!"-- the kid grasps your forearm and says, "Mr. Jones, Welcome to Xanadu!"

The stand is what you dial up from your home computer; it is where you sign publication chits, change your password.

Perhaps you have business meetings there, or drop by to explore hyper-art while you have lunch. (You order from your screen, and a Conductor (or "bystander") brings your food on a cart. Gracious.

There are no coin slots. Addfares devices, like those in the D.C. etro, recharge your use of the machines. Xanadu credit cards allow services to be charged. The silver allows you to charge meals as well, the gold Xanadu card charges anything.

The Xanadu Station is not impersonal; people stand face-to-face, without counters. It is installed throughout with quality video monitors and doubleseats. You may rent a computer or bring in your own Apple, GIGI, or what have you, with your own encoding devices embedded to taste.

There is a comparatively noisy social area near the front where the true Xanies hang out, with high-power E&S equipment. Quieter carrels are to be found in the rest of the building.

The buildings are designed for expansion. The foundations are overbuilt; the central area will be permanent; the rest will be built from a modular hexagonal building-kit system with furniture and monitors built into wall panels.

The outer angles of the castle-like exterior will therefore be hexagonal pleatings. They will be faced with semisilvered twin glass, allowing external view without bringing in too much light. The building will expand simply by adding uprights and interchanging panels.

A geodesic dome, starting as a very small arc, grows and is rearranged as the whole configuration expands. There is a large central room with a terraced cone-shape under the dome, growing with it. On its terraces are caré and lounge areas, as well as more and more screen areas.

The raucous, brainless atmosphere of the videogame parlor is not here. People typically come for an hour or more, relax and concentrate here, often in pairs.

Every stand has of course a main feeder machine (VAX 780, BBN C machine, or cost-effective alternatives). It is visible through glass, starkly lit. But the disks don't have to be seen, and should probably be underground to save space.

THE UNIFORMS

The uniforms have Nift. Jeans are standard, the rest changes periodically, as at McDonald's. The motif is silver, bug with other colors changing seasonally. The outfit is slightly wiggy and "futuristic," with a changing assortment of sashes, flyer's helmets, sommelier's chains, lightsabres, blasters and so on.

BUSINESS PLAN

Products and Services

Xanadu will run on any computer with a "C" compiler and an address space larger than 64k bytes. For example, a Motorola 68000, VAX/780, or an IBM 370 running the UNIX operating system.

XOC, Inc.

XOC will be selling licenses to use the Xanadu object code on a customers in-house computer. In addition we will be selling a front-end program of our own design for general documentation use. OEM licenses are also available to marketers of software for Xanadu and our front-end running as an integrated system. We will provide front-end design and development services by contract.

Project Xanadu, Inc.

Project Xanadu plans to operate an electronic library and publishing network for use by the public, providing Xanadu by the hour. It should be available soon ('82 ?). But we have said that before. Services will be provided nationwide via packet networks and locally at Silverstands. Please contact us for terms, availablilty, our demostration schedule, documentation and any other questions you might have.

Software Authors

We desire independant software authors and organizations to build front-end programs of (text, graphics, audio, video) all designs for any computer especially popular 8 and 16 bit microcomputers (Apple, TRS-80, PET, Atari, Heath, Xerox, IBM, Z8000, 8086, MC68000).

Development Status

The current version of Xanadu (11/10/81) is a single-user version. It works but is still being tested. Conversion to multi-user and other minor configuration changes are necessary to deliver to a test site. We expect this work to be completed in the next few months.

P.O. Box 7615,
Ann Arbor, MI 48104 USA, Phone (313) 663-3637

Written by Brian Wanty at XOC, Inc. 11/81

TOWARD A SUBCULTURE OF INTELLECT

Intellectualism is not a body of knowledge or a subculture, but a questioning, observing, hypothesizing outlook.

There are no intellectual subjects. For someone used to learning, to grabbing vocabulary and ideas, the elements of a new subject can come quickly. The more diagrams you have seen, the more words you know, the more theories you have heard, the more easily you can grasp the next one and assimilate it to the snowball of ideas already rolling through your head.

In an era of school-induced stupor, punch-and-judy news and video narcosis, we hope the Xanadu System will encourage depth and a never-ending procession of new insights. It will not be just for the pompous Establishment test-takers of Mensa or Magister Ludi-style nexocrats, but for whoop-te-do enthusiasts who enjoy sharing their sophistications. It's a fast lane for ZIps, a picture-book for Bozos, and for night people as well as day.

---

* CHANGING THE SABBATH

It is important and customary for a cult to change the sabbath. That way the followers see proportionately more of each other and feel set apart.

Well, the week is sinking toward uniformity no matter what, and a lot of these will be Night People, so our approach is a little different: to change the time-system.

When LaGrange and others created the metric system, that triumph of rationality scorned by yokels two centuries later, the day was of course also divided in ten. But after the French Revolution there was a squabbling as to whether it should be 10 units for the former 24 hours or 10 for the AM and 10 for the PM, and the whole thing fell apart, to our everlasting loss; we are stuck with the Babylonian time system and ungodly conversions between such things as frame-rates and minutes.

Well, the Xanadu system will undoubtedly be used for international teleconferencing, and so the time-zone problems (unfortunately also built around a system of 24) will come in. So we intend to adopt a timing-system to facilitate global and off-planet chitchat, which at the same time provides fun and games and a zippy sense of participation. (cont'd. next page)

We think we can build a new subculture of intellect, intellect in a new and enthusiastic style--more like the sci-fi subculture than Academia.

Here is a bunch of people who are paid to sit around and make things interesting for you. A national corps of peripatetic smarties plus the local bright kids are the Xanadu Conductors. The local kids (and others) run the stand; the national Xanadu Corps moves around, shares insights and explorations, do demos constantly for each other and for the local kids at a given stand, showing them what's new and what they've recently discovered in different subject realms on the system.

The people who hang around the stands will be an active subculture -- bright, verbal, interested in everything. Just as in music circles it is customary to know about symphonies and in sports circles it is customary to know about scores, in Xanadu circles it is customary to know about everything: to exchange interesting anecdotes, remarkable facts, extraordinary interconnections-- and converse with a Xanadu screen near at hand for reference. They'll be like trivia freaks or D&D players-- but with a more generalist outlook.

There'll be festivals and events: Hypercons, Kublacons and Front-End Functions, Footnote Festivals and Intertwingularity Expos. There'll probably be an argot; perhaps Porlock and Rosebud codes, something like the ten-codes of CB radio.

(cont. from prev. page)

Begin with the time in Greenwich, England. Midnight=0, noon=.5, 9 PM=.75. amd so on. This notation we may call GAD, Greenwich Absolute Decimal. Prefix this with the day of the year (e.g, 1 Feb=32), and you have absolute day and date, such as 364.99, New Year's Tooting. This is Xanadu Universal Time or XUT (alors). *It is the same everywhere*. Including (uh-oh) the date *as of Greenwich*.

That last part does throw a pall on things, so it must be adjusted with a time-zone offset. That is a number, positive or negative, permanent to your location, which gives you the correct date and sun time where you are, in decimal. Conversion is of course a front-end function.

This is the time as most prominently shown at the Silverstands. (Normal Babylonian time may be read from a cuckoo clock here and there.) Anyone wishing to market watches, clocks, etc., embracing these principles will get the use of our trademarks, etc., for a nominal consideration.

Please note that specifications are not finalized and do not represent a studied or unanimous decision by the Xanadu group.

A CULT?

Yes, call it a cult if that makes the idea clearer; but a secular cult of fascination with ideas, ready to wrap itself around the new or the old.

Cults do not just happen. They are constructed. If they become successful, it is through careful planning and insight about what works.*

WHAT WORKS?

This cult offers a social system with its own status ladder (highest are the travelling generalists), a promise of "education" to reassure parents with-- it's better than pinball, right? (And cheaper per hour.)

3/11

FREEDOM IN OUR TIME AND BEYOND

NEW FORMS OF IMPRISONMENT

With the best inventions, new forms of restriction and imprisonment involving computer screens are turning up.

A reader can close a book or skip to the ending. In some new environments, such as Computer-Assisted Instruction, it is possible to fully trap the user so that he has no options whatever except what the planner intended. I submit that this is not our free tradition.

A whole new set of rules is about to be generated. You may be supposing you have freedoms that aren't there anymore. Consider wiretapping. Time was when a wiretapper sat in the basement with earphones under his fedora. No more.

Your phone now can be tapped by a person thousands of miles away who simply gives the proper commands, as beeps, to your ESS switching station. Or so I am told by my telephone-knowedgeable friends, and I believe them.

So in principle, if we ever get the Wrong Sort of Government, they can study your life from your telephone use like an open book. So that avenue is closed.

What, then, about your computer transmissions?-- and we know they will be a basic form of communication for individuals soon.

An ominous situation has already arisen. The U.S. government has given its approval to a system of encoding that you can buy on a chip. This system is called DES, Data Encryption Standard. Since you can prime it with a secret 56-bit "key," it is thought that running your transmissions through it acts like an electronic shredder, hopelessly garbling your transmissions except to the other party, who presumably knows the key you are transmitting with.

There is only one problem. The DES system stands impeached. Researhcers on both coasts have accused this method of being easily breakable by the National Security Agency, the government's decoding arm.

Another system of codes has been proposed that supposedly can't be broken by <u>any</u> extant computer in less than millions of years. This is the RSA code, developed by Rivest, Shamir and Adelman of MIT. It has several remarkable properties, among them being the ability to exchange unbreakable messages between strangers who have not had a chance to swap code keys; the ability to co-sign electronic documents that anyone can read and know you signed; and more.

This may seem faroff to the average reader-- perhaps as faroff as television seemed in the nineteen-thirties. Then consider the following:

If you are not careful, some government may be able to read your private computer documents in the future at any time.

THE MINISTRY OF TRUTH,
THE THOUGHT POLICE,
AND YOU

1. The MOT

In Orwell's 1984 the Ministry of Truth told lies and changed history. This was done by judicious snipping, disposal and replacement of paper.

As documents go electronic, however, no longer need paper be involved. A reference article, say, in an encyclopedia can be changed simply by storing another one in its place--and poof! history is changed.

There is no typography or watermark to check. Characters sent on the wire are all alike. But if the right sort of encoding is used it will be some sort of deterrent to the nefariousness of governments of the future.

2. The TP

The Thought Police couldn't really read your mind, but they knew enough psychology to have good suspicions.

Tomorrow's real-life Thought Police will have detailed access to a huge number of incidental records about your life, from banks and auto registration and so on-- instantly investigable.

But I don't think you want them to know what you store, or what you read. So let's talk about it.

Strong codes are available for storage (like RSA). That takes care of making your private documents safe (-- except where do you hide the code? -- another problem).

But let's assume that a dictatorship of stealth is seizing the stations, one at a time. That essentially gives the TP access to what you've asked for as it passes through, with you as the designated recipient.

A good deal of concerned research is going into this problem from many points of view, and we can't really get into it here; but many solutions might be possible. One would be to somehow reveal at each station only what its next pass-through station would be, with your final name only unwrapped in the last station.

THINK FAST

These problems are real and present. I did not create them. They were here waiting for us. They will come to be very important. The way to approach these issues, I believe, is not sit in a corner and tremble, like a rabbit in a tiger cage hoping it won't be eaten, but to run between the legs of the beast before it fully wakes up.

CIVILIZATION AND ITS DISK-CONTENTS

There are many bombs and fires to come, and Deep Digital-- deep rock, deep space-- is really the only long-haul solution.

The system this book proposes is a generalized and self-networking structure that can eventually be put in deep rock and deep space-- not the unique solution, but a method. We seek to provide a general method of storage and access that can be put in Iron Mountain or the asteroid belt.

Digital archiving can include text, paintings (stored as detailed photographs), sculpture, architecture, whatever.

We do not propose to be the guardians. Just to try to help.

TECHNICAL CHAPTERS

THE ONLY WAY IT COULD WORK

TUNING THE SYSTEM

THE TRADEMARKS

TUMBLING THROUGH THE DOCUVERSE

THE PROTOCOLS

THE ONLY WAY IT COULD WORK

THE ONLY WAY IT CAN POSSIBLY BE DONE

Some conventional methods, such as B-trees, permit rapid insertion and deletion in large structures. The slowdown as structure grows is *logarithmic*.

The ideas promoted in this book could not possibly be contemplated unless methods of storage, editing and linking could be found which *all*, singly and in combination, deteriorated in performance as a logarithmic function of the size of a document and the size of the docuverse. We believe we have achieved this. As in other dynamic-function problems, analytic proof is not possible, so this is an empirical question to be proven or disproven.

TUNING THE SYSTEM

TUNINGS

The system's design is a unified whole, but we may think of it as a combination of structures: the basic conceptual structure, plus a technical structure which makes it possible, and a contractual structure which makes it possible for people to use it confidently. These aspects taken together make a unified design. Because the conceptual structure required very fast lookup within a tightly organized but large linked system, we had to develop a particular technical structure; and because the conceptual structure expects participants to behave in certain ways, these are embraced in the contract offered to users. These provisions are necessary for the orderly and confident use of published material by many people.

We are concerned with the balance of customer incentives to help foster our overall goals. In the coalescing final design of the system, contracts, categories of service and pricing are all subject to reconsideration. We need to study possible cost functions for reducing possible Babel; or for cutting less-recent accessibilities in order to be practical.

The system has two business commandments, viz.:

1. EVERYBODY MAKES MONEY: there exist many opportunities for profitable participation.

2. ALL SERVICES MUST BE SELF-SUPPORTING.

The following discussions will investigate ramifications of the latter premise.

ROYALTIES FOR WHAT EXACTLY?

Granted that royalties should be exactly proportional to something, what should that be?

If we make it "transmissions," some paradoxes surface. For instance, if you the user have a fancy computer, you and your program may request many transmissions that are used little or not at all-- while certain materials, already transmitted, stay on your screen.

Fairness would suggest that the material on the screen, not what goes over the wire, should be the royalty divisor. This would require certain back-reporting by the front end and may be a can of worms, but it certainly has elements of fairness.

We have considered schemes for getting reports back from the user systems-- optional with the user-- stating, on an honor-system basis, where the royalty-fragments should go that are not measurable at our end as transmissions. But this may be too much flex; and many clowns would award the royalties to their own published works. So there is a case against this just on the basis of straightforwardness.

Or consider the user who has a low rate of transmissions, say 50% of the channel capacity if he chose to ask for fetches at full rate. Should 50% of the royalty go to the authors he used, and the rest to the author's fund?

BOUNCE-THROUGH ROYALTIES

When you use somebody's directory, you bounce through their specification to another document. What royalty goes where? (We want to encourage the creation of directories, so these authors should be rewarded.)

However, we also want to keep royalty a fixed rate.

One solution: transmit the full address of the desired document, which would yield a royalty proportional to the address length to the directory owner. He could even increase the incentive by increasing artificially the length of this transmission. But this reduces the capacity of the user's system by slowing him down.

MARKET-PRICING CONTROVERSY: FIXED VS. VARIABLE

There are two schools of thought with respect to the pricing of these services. Surely the amount transmitted should not vary the price, since that would discourage high mental rates-- not what we want at all.

One school of thought has it that certain flat, predictable charges-- such as ten dollars an hour or two dollars an hour-- depending on class of service-- are the best way to go. We can call this "smorgasbord" pricing-- one price for all. It has the special advantage of avoiding hanky-panky in the accounting programs, which can then have conspicuous checksums. Further, the user can predict his overall expenses nicely. Perhaps most important, a uniform royalty for all authors and documents is also desirable because this means there is no pretext for the system's keeping track of who reads what.

Therefore, just as the postoffice subsidized the outlying stations on the basis of profits from the easy parts in the interests of uniform service at a uniform price, so might we.

Classical economics, however, suggests a more buoyant pricing mechanism, varying with time or system load-- "level-seeking," allowing market factors to enter in in a useful fashion.

1. <u>Author Variations</u>.
One such market factor would be to allow authors to set their own royalties-- very high, if they wanted. For now, in Balance I we have opted not to feature this.

2. <u>Slack-time price float</u>.
In this view, unused capacity should seek a "spot" market price, selling for less in short, or interruptible blocks.

3. <u>Market price of disk</u>.
In later stages, allowing rental of disk to be distributed among various vendors, with some market-pricing mechanism, is not out of the question.

(Variant proposals to hold costs constant by <u>slowing down service</u> have been proposed, as a method of allowing the pricing mechanism to enter the situation while maintaining constant cost-per-hour and, e.g., charging higher royalties for materials for a specific source. On the positive side, this allows pricing dynamics to operate and might allow users to "break through" to full performance at a higher cost. On the negative side, it is philosophically most disagreeable.)

COST/TIME TRADEOFF

Cost and time are often a continuum. On our system, various areas of performance can be slowed down at lesser cost. In both behind-the-scenes and up-front ways, cost/time playoffs are important options in the tuning of the system.

THE RESOURCE UNIT

Users can ask for the moon and stars simultaneously. While early versions of the system will merely fetch what is asked for on a simple queuing basis, more sophisticated service algorithms will have to ration resources.

The Resource Unit (RU) then becomes a basic internal unit of software accounting, dividing the system's effort on your behalf. A standard customer gets one RU. (Priority customers might get more.)

If one entity is called for, the search for it proceeds with a force of one RU. If two entities are called for simultaneously, each gets 1/2 RU; and so on. RUs are divided as requests fan out.

It is easy to see why this is necessary. The request fanout can easily become astronomical, which is <u>all right</u>; the problem is to find an orderly basis for servicing MIRVed searches (Multiple Independent Reading Virtuality). The divisible Resource Unit keeps the overall Systems Effort equal to unity rather than inflating in combinatorial explosion.

ADVERTISING

The system does not discriminate in any way among "types" of document by content. Advertising is thus perforce allowed.

However, suggestions that advertising can somehow pay for generalized use of the system, as with TV and magazines, have pitfalls. Specifically, there is no foreseeable way to find out what is actually being <u>shown</u> on a screen; thus advertising could be automatically screened out in many ways, defeating the usefulness of it. So it is not clear that advertising subsidy is feasible.

ARCHIVING

It is mandatory that all sections of the service be profitable. This puts the question of "archiving" on a curious basis. Here the trade-off between cost and time comes into sharpest relief.

DISK STORAGE, on a unifying hypertext system, can be effectively instantaneous. But disk storage can last as long as somebody pays for it. After that you go to tape.

THE TAPE PROBLEM

What is on tape takes longer to get to. And bringing in materials for tape takes lots of time-- time that can worsen drastically as demand escalates. (In the degenerate case of queuing, of course, we go to repeated Grand Passes of the corpus.)

Immense automatic tape systems are available, with little locomotives that go to the appropriate rack of tapes, pluck the one desired, slurp it in and put it back. But the maintenance of a large-scale tape library is expensive-- though less, of course, than disk per unit stored.

TWO SCHOOLS OF THOUGHT ABOUT TAPE

One view is that disk and tape should be a unified whole, with all that is on tape, an indefinitely expanding bundle, united to what is on disk; though subject to unpredictable delays.

However, this creates financial difficulties. Most of what stays on tape is no longer being paid for or referred to. Storing the tape is cheap; hooking it up is expensive. Finding a viable economic arrangement is the key problem.

One solution, and a probable one, is simply a surcharge for tape fetches.

(This in addition to the delay, which is an implicit charge.)

ARCHIVING ARRANGEMENTS

There may, after a point, have to be a charge for long-term tape storage-- or for guaranteeing it. Our provisional solution is to define service arrangements, and the organizations to maintain them, rather like "perpetual care" mortuary arrangements. This will be discussed later.

BALANCE I

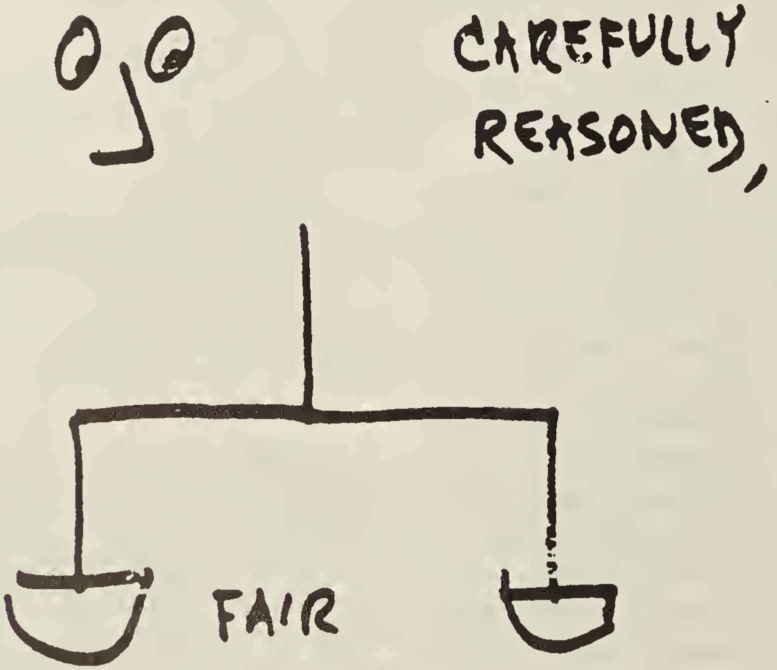
The overall scheme of incentives, this particular tuning, I call Balance I. It consists of the following provisions:

Two simple categories of privacy (published and private-- with private materials recallable, published requiring 6-month depublication notice. "Privashing" recallable, unlimited distribution, no royalty.

Fixed royalty, 5% of hourly charge, so that the computation of royalty is simpl e (to avoid hankypanky in the accounting).

Fixed charge by hour, not by amount transmitted. (We want to encourage people to read a lot, not to reduce their intake!)

Note that this arrangement is fair, orderly and simple. And these seem to me very important features.



If documents are used that have nobody for a royalty to go to, such as Shakespeare (or the use of your own private documents), the same amount goes to the Authors' Fund. A blue-ribbon panel assigns these proceeds to such purposes as typing in Persian poets or subsidizing writers.

Royalty to publishers of directories is proportional to the coordinate-material they transmit.

THE QUESTION OF SPECIAL SERVICES

In addition to the standard services, we could get into additional speedups within our philosophy (look-ahead, faster copies of specific versions). Or we could imitate other services by adding admittedly-useful features that they have and we don't.

Consider _scans_. Many retrieval systems have fulltext scans for keywords. In doing that we are intrinsically less efficient than anyone else, since they store blocks and we store fragments. However, by storing _concordances_ we could effectively achieve the same searches. Ideally these would be concordances built concurrently with input typing. These would enlarge storage for a given document 1/5 to 1/3, however.

But we must always remember that our specialties are the rapid delivery of linked and windowing documents, and the assimilation and storage of changes. If we go far afield from this, our system could lose its power and ideals.

THE TRADEMARKS

The following are the trade and service marks of the system described in this book, by which we distinguish our information services and products from others.

"XANADU^{tm}" to denote all our information services and products.

"SilverStands^{tm}" to denote the stations at which service will be provided.

"FREND^{tm}" as an authorized and approved FRont-END program or console.

"XANAMAIL^{tm}" to denote personal message services.

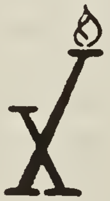
"XANACARE^{tm}" to denote arrangements for guaranteed long-germ storage.

"XANADOODLE" for computer graphic systems.

The slogans
"Lightning Literature^{tm}"
"The World of You^{tm}"
"The Wings of Mind^{tm}"
"Anything Instantly^{tm}"

The following cartoon characters:
PORLOCK^{tm}
ROSEBUD^{tm}
XAN MAN^{tm}
The MARGINALIEN^{tm}
and
THE HOBGOBLIN OF LITTLE MINDS.^{tm}

And finally that X-ternal Device,
The Eternal Flaming X
in all its variants.


tm

TUMBLING THROUGH THE DOCUVERSE

Our kingdom is already twice the size of Spain,
and every day we drift makes it bigger.

The Kaiser
in Herzog's film
Aguirre, The Wrath of God

Besides the actual contents of our system-- text, graphical data, and other notations representing things people want to look at and manipulate -- the system must keep track of a lot of numbers. These are the internal numbers that are used for counts and pointers, and the overall scheme of where things are and how to get to them. They are integers.* Some of these numbers have to be very very big. Others (in fact most of them) are small.

Our universe of documents (or docuverse) is potentially very large, and will grow unpredictably. Numerical addresses in our system can therefore grow very large. But they must also work with small increments and offsets. Designing the address space and notational representation is therefore crucial and difficult.

It is not obvious-- it was certainly not obvious to us at the outset -- how to specify such a universe in any tractable form, with an indexing scheme that can possibly grow very large and still retain any cogent manipulability when we deal with the nitty-gritty small increments of changes within a given document.

One assumption would treat the docuverse as a large integer domain, sparsely occupied by assigned document addresses. That way lies madness: it would mean unoccupied areas would use up many, many precious bits.

* Except where floating-point and trigonometric functions are used for certain proprietary algorithms.

We drew some inspiration from the Dewey Decimal system, which, despite its faults, does not waste a lot of space on empty characters. This leads to insights about forking numbers, which we have developed in an unusual way.

There are many kinds of numbers and notations for them. While it is customary in computer work to use several kinds of numbers (integers and floating-point numbers of different lengths, ASCII and BCD decimal trains), we use none of these in our current system design. For the interested reader, the types of numbers we have chosen to use are an interesting exercise in notational engineering.

Our solution has two parts. One is to use an accordion-like integer notation whose numbers are very short in representation when small, and as large as they need to be when big.

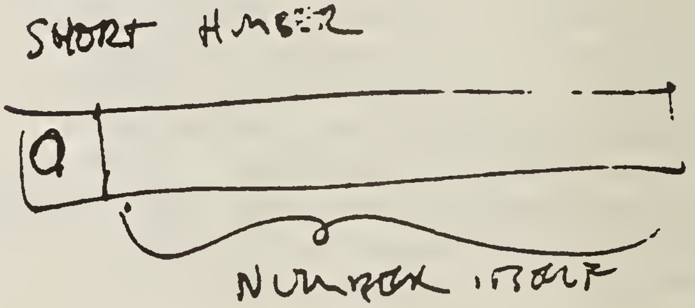
The second part of the solution is to define an accordion-like master address space, potentially very very large, which includes notational provisions for the complex relations between documents, their forebears, their owners, their locale, and the expansion of the network itself.

HUMBERS
(Variable-Length Binary Integers)

Humber stands for "Huffman-encoded number," which (strictly speaking) it is not; so if you prefer it stands for "humungous number."

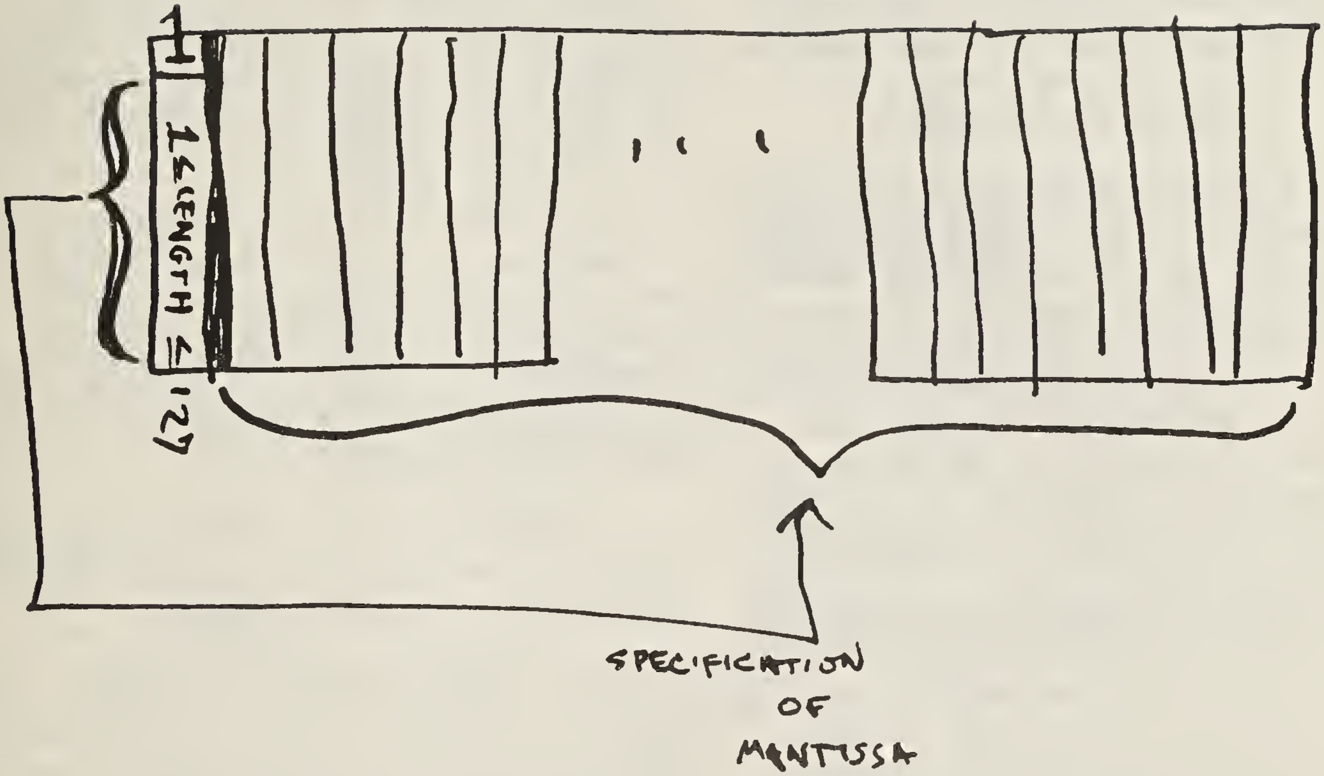
Consider a byte.

The first bit signals whether the number is <u>complete in this byte</u>. If this bit is unset, or zero, the remaining seven bits hold the number itself (Ø to 127), and the entire number is stored in the one byte.



If the Completeness bit is set, or one, that means the remaining bits of this byte specify the length, in bytes, of the number, in binary. Thus the humber may range up to $2^{127 \times 8}$, a number larger than needed very soon.

LONG HUMBER



Note, then, several advantages of this scheme. Small incremental humbers are one byte long. But very large humbers adhere to the same format; only one set of "humber arithmetic" routines is necessary.

It will be noted that these numbers occupy no more space than they need; they are short most of the time (when needed for small incrementation) and stretch out whenever needed without any change in the generalized manipulation routines. No more than seven bits are wasted in the length of the mantissa, and there is only the one-byte overhead of the specifier.

The Containment bit is zero if the actual number is within the byte, 1 if it is not; this choice makes an all-zero byte a true zero (a fact which will be seen to be a useful choice for the tumbler mechanism, below).

The Master Address Space:
TUMBLERS
Forking Multipart Integer Vectors, with Carry

The larger scheme for addressing in the docuverse, our present one, employs a multi-part number with some rather remarkable features. It is intended to keep track of hereditary successions of various kinds, while reducing the overall indexing manipulations to tractable arithmetic form. We call it a <u>tumbler</u>.

We chose the word "tumbler" partly because it sounded like "number" and "humber," and partly because of its curious relation to the rotary mechanisms of locks, which also slide with respect to one another, and are also called "tumblers."

The diabolical simplifying assumption that we have made is that <u>there is really only one document</u>.

A hypothesis built into the scheme is the notion that the number of compatible nodes will grow indefinitely but in hard-to-predict patterns.

Thus a node, or station, is seen as having ancestors and possible descendants. An account, too, and a document, all have possible descendants.

For instance, consider that you have written the twentieth document, #20, on a given node. Now you do a Version Fork, leaving you two versions which you choose to designate as separate versions, the original number being superseded.* Now these two are versions 20.1 and 20.2 (while the parent document, 20, in fact continues to hold most of the contents of both). Now suppose you do another version fork on 20.2. This yields 20.2.1 and 20.2.2-- and so on.

The entire tumbler works like that: nodes can spin off nodes, accounts can spin off accounts; nodes can spin off accounts and documents; and so on. Thus all numeration in the docuverse is comprised to a single mechanism.

The tumbler format is:

NODE NO.	ACCOUNT NO.	DOCUMENT NO.	VERSION NO.	POSITION in version

Each of these fields may have one or more parts.

Two different field separators are required. Presently we use the decimal point for hereditary junctures within a given section; and the number Ø (between decimal points) to separate the major sections. Thus a tumbler might look like:

Ø.Ø.Ø.7.3.2.Ø.335.896

(The fields missing between the first three zeroes show this to be an _incremental_ tumbler.)

A fuller specification of the tumbler is as follows:

H.Ø.H.Ø.H.Ø.H.Ø.H

where "H" is any hereditary multihumber (series of humbers representing hereditary segmentation by decimal points as described above).

Thus a large tumbler might look like this:

i...i.Ø.i...i.Ø.i...i.Ø.i...i.Ø.i...i

where "i" is any integer, represented how you like.

* Whether a parent version number continues on to evolve as one of the daughter versions is a user choice, and thus a front-end function.

(Note that we have skipped over the notational reconcilation between humbers and decimal points. If properly "understood," the decimal points can be simply left out.)

There is not time at the present writing to explain the rules of tumbler arithmetic as worked out by the group. Suffice it to say that tumbler addition is non-commutative (A+B does not equal B+A) and therefore there are strong and weak forms of subtraction (given A+B, both A-B and B-A).

It will be seen that the tumbler (and its associated routines of addition and subtraction) provides a master scheme for the full address space while handling increments and offsets -- whether local or very large-- with creditable brevity. These increments, and offsets, naturally, can cross the lines between nodes and accounts, documents and versions. So it's all really one big forking document.

GENERAL REMARKS

The docuverse is the <u>occupied</u> address-space. We do not waste numerical positions on what is not there. As with Dewey Decimal, conceptual holes do not become utterly inefficient notational holes.

Note that "time" is not included in the tumbler. This results in part from the interesting hypothesis that at some future time, document nodes may be in starships nearing the speed of light, so that their time records will not transform directly to those kept at stable locations. Time is kept track of differently.

Note also that our demonstration system uses standard integers as tumbler fields, not humbers. However, our coding is modulzrized in anticipation of this upgrade.

Humbers are the work of Roger Gregory, Mark Miller and Stuart Greene, done in the summer of 1979. While it went through many changes, and represents contributions by numerous members of the Xanadu troupe, the present form of tumbler was worked out by Mark S. Miller, with help from Roland King and Roger Gregory, in approximately June of 1980.

FEEL FREE

No proprietariety is asserted for humbler and tumbler methods or for their names; use them freely. (Indeed, they are a required part of the front ends.) However, the group would not mind a little credit now and then.

*** XANADU BACK END SPECIFICATIONS ***

THE PROTOCOLS

NOTE

THIS IS A PRELIMINARY SET OF SPECIFICATIONS

The first released version will be different in detail. Be sure to contact Project Xanadu (tm) before using them to write a front end.

WHATS'S IN THE SYSTEM -

Documents, characters, and links. Each is identified by a tumbler q.v. which is called its v[irtual] stream address, or vsa.

DOCUMENTS

A Xanadu document consists of 1) a name (its vsa), a tumbler assigned to the document by the system when the document is created; 2) text, a string of ASCII characters; and 3) links, which appear in the document by their names which are tumblers assinged to the system when they are created. The text and links together make up the contents of the document. The individual characters and links in the document are called things, or, collectively, stuff. Each thing in a document has an address, which is a tumbler. Normally all the test in the document will preceed all the links. Characters have addresses which begin with 1. and links have addresses which begin with 2., for example:

```
document # 1.0.1.0.23

          1.1 h
          1.2 e
          1.3 l
          1.4 l
          1.5 o
          2.1 1.0.1.0.2.1
          2.2 1.0.45.0.2.77
```

4/17

The portion of the document between addresses 1.0 and 2.0 is called the text space of the document; the portion with addresses greater than 2.0 is called the link space. Addresses 1.0 and 2.0 themselves are not used. A document must have a name. It may contain text or links or both, or it may be empty.

SPECS AND SPECSETS

A spec is defined as either a docrange or an vspec. A docrange is described by two tumblers, and refers to the set of documents with addresses between A and A′, inclusive, where A is the first tumbler and A′ is the sum of the first and the second. For example:

if these are all the documents between 1.0.1.0.1 and 1.0.1.0.5

```
1.0.1.0.1
1.0.1.0.2
1.0.1.0.3
1.0.1.0.3.1
1.0.1.0.3.1.1
1.0.1.0.3.2
1.0.1.0.4
1.0.1.0.5
```

then this docrange: 1.0.1.0.1 0.0.0.0.1

will produce this set of documents:

```
1.0.1.0.1
1.0.1.0.2
```

and this docrange: 1.0.1.0.3 0.0.0.0.1

will produce this set of documents:

```
1.0.1.0.3
1.0.1.0.3.1
1.0.1.0.3.1.1
1.0.1.0.3.2
1.0.1.0.4
```

this docrange: 1.0.1.0.3 0.0.0.0.0.0.1

will produce this set of documents:

```
1.0.1.0.3
1.0.1.0.3.1
1.0.1.0.3.1.1
```

An vspec is defined as a docid followed by a spanset, which is a list of spans. A span is a pair of tumblers, an origin and a width, which defines a region of a document. The origin is the address of the first thing in the region, and the sum of the origin and the width the address of the first thing past the end of the region. For example:

```
    docid  1.0.1.0.24

           origin 1.4
           width  0.3
```

refers to the 4th through the 6th characters in document 1.0.1.0.24

```
    docid  1.0.1.0.24

           origin 2.1
           width  0.4

           origin 2.10
           width  0.1
```

refers to the 1st through the 4th and the 10th links in document 1.0.1.0.24

```
    docid  1.0.1.0.24

           origin 1.1
           width  1.10
```

refers to all the text and the first 10 links in document 1.0.1.0.24 (Mixed text and link spans won't work in the current implementation.)

A specset is a list of docranges and vspecs.

4/19

FORMAL DEFINITIONS OF TERMS -

```
<tumbler> := <texp> <tumblerdigit>* <wdelim>
<texp> := <integer>
<tumblerdigit> := <tdelim> <integer>
<tdelim> := '.'

<wdelim> := '\n'

<docid> := <tumbler>
<docvsa> := <tumbler>
<linkid> := <tumbler>

<span> := <tumbler> <tumbler>
<spanset> := <nspans> <span>*
<nspans> = <integer> <wdelim>

<specset> := <nspecs> <spec>*
<nspecs> := <integer> <wdelim>
<spec> := <specid> { <vspec> | <span> }
<specid> := { 's' | 'v' } <wdelim>   /* v for vspec, s for span (docrange) */
<vspec> := <docid> <spanset>

<textset> := <ntexts> <text>*
<ntexts> := <integer> <wdelim>
<text> := <textflag> <nchars> <char>* ?<wdelim>
<textflag> := 't'
<nchars> := <integer> <wdelim>
<docset> := <ndocs> <docid>*
<ndocs> := <integer> <wdelim>

<linkset> := <nlinks> <linkid>*
<nlinks> := <integer> <wdelim>
```

4|20

THE REQUESTS -

```
<request> := <    INSERT
              || CREATENEWDOCUMENT
              || DELETEVSPAN
              || REARRANGE
              || COPY
              || CREATENEWVERSION
              || MAKELINK
              || RETRIEVEV
              || RETRIEVEDOCVSPAN
              || RETRIEVEDOCVSPANSET
              || FINDDOCSCONTAINING
              || FINDLINKSFROMTO
              || FINDNUMOFLINKSFROMTO
              || FINDNEXTNLINKSFROMTO
              || APPEND
              || FINDDOCSCONTAINING   >
```

```
INSERT := <insertrequest> <docid> <docvsa> <textset>
returns <insertrequest>

  <insertrequest> := 0 <wdelim>
```

puts given text in document at given address, vstream addresses of following characters, if any, are increased by length of inserted text.

```
CREATENEDOCUMENT := <createdocrequest>
returns <createdocrequest> <docid>

  <createdocrequest> := 11 <wdelim>
```

creates an empty document, returns docid of new document.

```
DELETEVSPAN <deleterequest> <docid> <span>
returns <deleterequest>

  <deleterequest> := 12 <wdelim>
```

removes given spans from given document.

4/21

REARRANGE <rearrangerequest> <docid> <cutset>
returns <rearrangerequest>

```
<rearrangerequest> := 3 <wdelim>
<cutset> := <ncuts> <docvsa>*
<ncuts> := <integer> <wdelim>               /* ncuts = 3 or 4 */
```

cutset consists of 3 or 4 vsas within given document -- in 3-cut case material between 1st and 2nd cuts is interchanged with that between 2nd and 3rd cuts, in 4-cut case material between 1st and 2nd is interchanged with that between 3rd and 4th.

COPY <copyrequest> <docid> <docvsa> <specset>
returns <copyrequest>

```
<copyrequest> := 2 <wdelim>
```

material determined by sourcespecset (given vspans of given documents) is copied to document determined by <docid> at address determined by <docvsa>.

CREATENEWVERSION <createversionrequest>
returns <createversionrequest> <docid>

```
<createversionrequest> := 13 <wordelim>
```

creates new document with contents of given document, returns docid of new document.

MAKELINK <makelinkrequest> <docid> <docvsa> <fromset> <toset>
returns <makelinkrequest> <linkid>

```
<makelinkrequest> := 4 <wdelim>
<fromset> := <toset> := <specset>
```

creates link in given document at given address from fromset to toset. returns linkid of link made.

```
RETRIEVEV  <retrieverequest> <specset>
returns <retrieverequest> <vstuffset>

  <retrieverequest> := 5 <wdelim>
  <vstuffset> := <nthings> <vthing>*
  <nthings> := <integer> <wdelim>
  <vthing> := <text> | <linkid>
```

returns material (text and links) determined by specset.

```
RETRIEVEDOCVSPAN <docvspanrequest> <docid>
returns <docvspanrequest> <span>

  <docvspanrequest> := 14 <wdelim>
```

returns a span describing the origin and extent of the vstream of the given document

```
RETRIEVEDOCVSPANSET <docvspansetrequest> <docid>
returns <docvspansetrequest> <spanset>

  <docvspansetrequest> := 1 <wdelim>
```

returns a spanset determining all sections of the vstream of the given document corresponding to distinct ispans

```
FINDDOCSCONTAINING <docscontainingrequest> <specset>
returns <docscontainingrequest> <docset>

  <docscontainingrequest> := 22 <wdelim>
```

returns a list of all documents containing any of the material determined by the given vspecset

```
FINDLINKSFROMTO <linksrequest> <homeset> <fromset> <toset>
returns <linksrequest> <linkset>

  <linksrequest> := 7 <wdelim>
  <fromset> := <toset> := <homeset> := <specset>
```

returns a list of all links which are (1) in homeset, (2) from all or any part of fromset and (3) to all or any part of toset

```
FINDNUMOFLINKSFROMTO <nlinksrequest> <homeset> <fromset> <toset>
returns <nlinksrequest> <nlinks>

  <nlinksrequest> := 6 <wdelim>
  returns the number of links which are (1) in homeset, (2) from all or
  any part of fromset and (3) to all or any part of toset


FINDNEXTNLINKSFROMTO <nextnlinksrequest> <specset> <fromset> <toset> <linkid> <nl
inks>
returns <nextnlinksrequest> <linkset>

  <nextnlinksrequest> := 8 <wdelim>

  returns a list of all links which are (1) in the list determined
  by homeset, fromset, and toset, as in FINDLINKSFROMTO, (2) past the link
  given by <linkisa> on that list and (3) no more than <n> items past
  that link on that list


APPEND <appendrequest> <textset> <docid>
returns <appendrequest>

  <appendrequest> := 19 <wdelim>

  appends given text onto the end of the text space of the given document.


FINDDOCSCONTAINING <docscontainingrequest> <specset>
returns <docset>

  <docscontainingrequest> := 22 <wdelim>

  returns list of all documents containing any portion of the material
  included in the given specset.
```

4/24

ABOUT THE AUTHOR

Ted Nelson is a rogue intellectual, social critic, and designer of interactive computer systems for our world of tomorrow. He has been designing computer text systems on his own since 1960, and may thus be considered one of the inventors of word processing.

Nelson holds degrees from Swarthmore and Harvard, and has taught in a variety of university departments. He has consulted to Bell and CBS Labs, IBM, the CIA, the NSF, and Walt Disney Productions. For the year 1980 he was the editor of *Creative Computing* magazine. His book *Computer Lib* is in its seventh printing, and his book *The Home Computer Revolution* has been translated into Japanese.

Nelson's ideas have a wide underground following in the computer field. His maverick point of view stresses the art and conceptual integration of screen graphics and other responding environments.

A dynamic orator, he was an invited speaker at the Eighth World Computer Congress, giving his talk, "Replacing the Printed Word, in Tokyo and Melbourne.

This is a report on his life's work, the Xanadu Hypertext System, a scheme for instantaneous publishing and archiving with very broad implications-- designed for "children, researchers and heads of state."

He publishes his own books because he does not like to compromise.

EPILOG

FATE ACCOMPLI

This is a Caper story-- a beckoning dream at the far edge of possibility that has been too good to let go of, and just too far away to reach, for half my life. The intrepid little group-- my comedy burglar team-- has gotten far closer to this dream than any sane person could have thought possible.

Though everything has seemd to block our way, on balance the Fates have been very much with us: laying down a trail of crumbs, as it were, through a very strange forest to a very unusual place. We propose to build a palace here and let you all inside.

Whether the ground will hold, what beasts and trolls may assail us here, all remain to be ascertained. We have done our best and will continue to do so.

We bring banners. We have held to ideals created long ago, in different times and places, the very best ideals we could find. We have carried these banners unstained to this new place, we now plant them and hope to see them floating in the wind. But it is dark and quiet and lonely here, and not yet dawn.

Now it is for you the reader to examine this place and say where, if anywhere, you would rather be. We hope you share our sense of urgency and of history. The choices are fewer than you might have thought, and perhaps they need to be made quickly. Good luck to you, and to us all.