

Original Publication

Excerpts from chapter 2 of *Literary Machines*. Sausalito, Calif.: Mindful Press, 1981.

# From *Literary Machines*

## Proposal for a Universal Electronic Publishing System and Archive

Theodor H. Nelson

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

### 1 An Electronic Literary System

#### 2 What is Literature?

#### 3 A True Storage System for Text and Other Evolving Structures

#### 4 A Linking System for Text and Other Data

#### 5 The Document Convention

#### 6 Compound Documents

443

### 7 Electronic Publishing:

#### Making the Literary System Universal

#### 8 Distribution and Networking

#### 9 Vital Issues

### 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.

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.

## 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

### 1

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 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 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.

down in what you usually do and think about when you don't use a computer.

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 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 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 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.

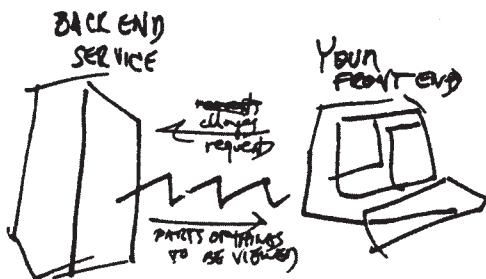
Storage is fundamental. What you store should be the basic structure of the information you are dealing with—not some tricky arrangement that is carefully matched to some set of programs or hardware. (How you will look at this world when it is spreadeagled on your screen is your own business: you control it by your choice of screen hardware, by your choice of 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, now 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.

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.

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

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 *that* 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.

Suggestions are welcome—if you are sure you understand this design first.

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.

## 2 What Is Literature?

Literature is an ongoing system of interconnecting documents.

### 2

#### 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.

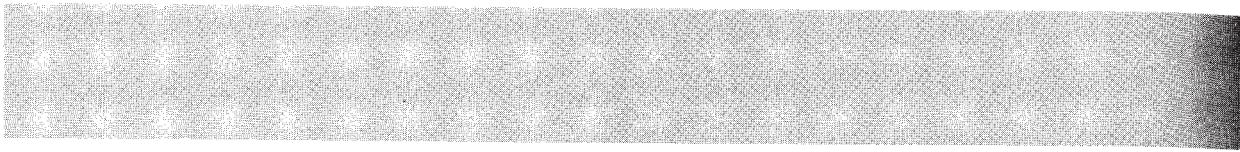
A person reads an article. He says to himself, "Where have I seen something like that before? Oh yes—" and the previous connection is brought mentally into play.

445

*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 his 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 argues 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, another 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.

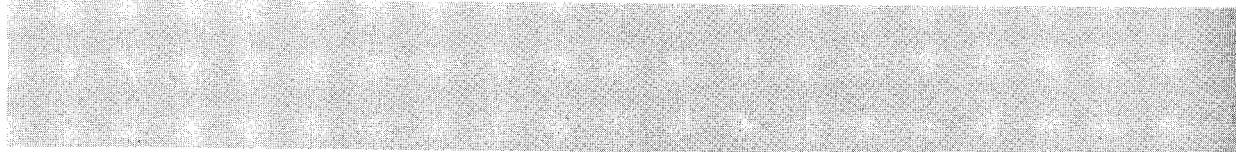
In other words, even though in every field there is an ever-changing flux of emphasis and perspective and distortion, and an ever-changing fashion in content and approach, the ongoing mechanism of written and published text furnishes a flexible vehicle for this change, continually adapting. Linkage structure 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.



### 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.

3

#### Prologue: Making Extra Copies All the Time

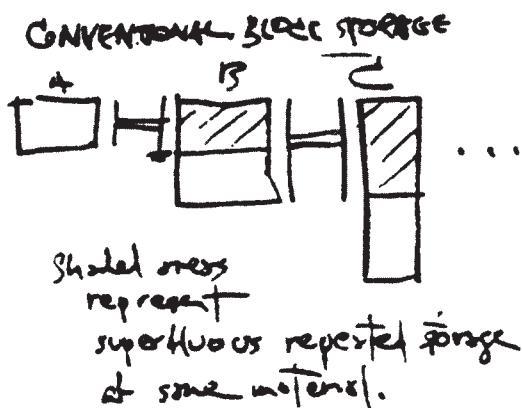
In most computer applications (such as the layman's newest game, "word processing"), it is often necessary to keep repeated copies. This frequent and disagreeable problem has several purposes.

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, *backtrack*, 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 right. 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



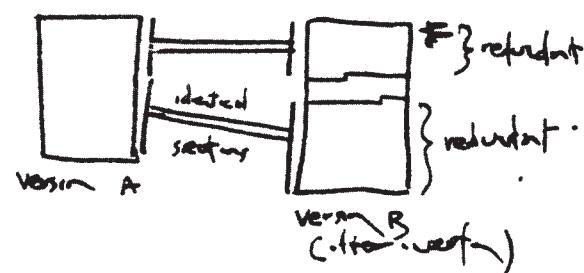
\*Including block-transfer circuitry, most display buffering, and the disk routines supplied with conventional operating systems.

but turns out to have remarkable power later on.

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

small changes and additions, then you are repeatedly storing the same material, redundantly.

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.

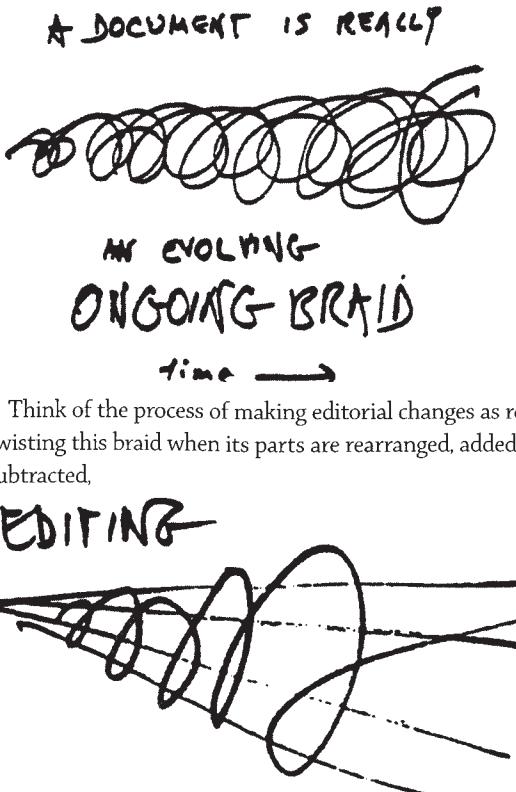
Think of it this way. An evolving document is really not just a block of text characters, Scrabble™ tiles all in a row; it is an ongoing changing flux. Think of its progress through time as a sort of braid or vortex.

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

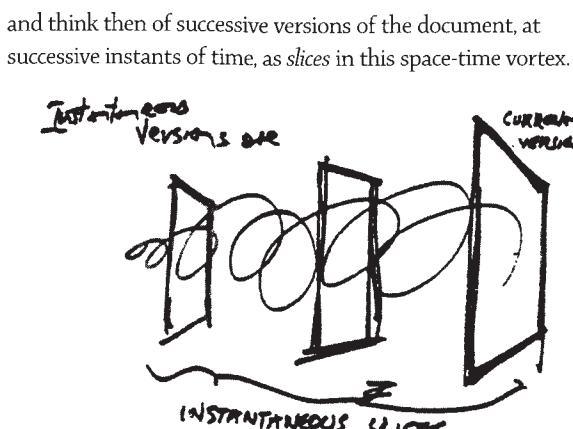
The true storage of text should be in a system that stores each change and fragment individually, assimilating each change as it arrives, but keeping the former changes; integrating them all by means

of an indexing method that allows any previous instant to be reconstructed.

This can be done efficiently if the user is reading from a computer screen; since you can set up the

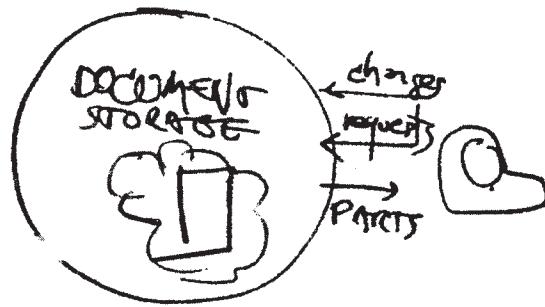


448



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.



#### 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.\*

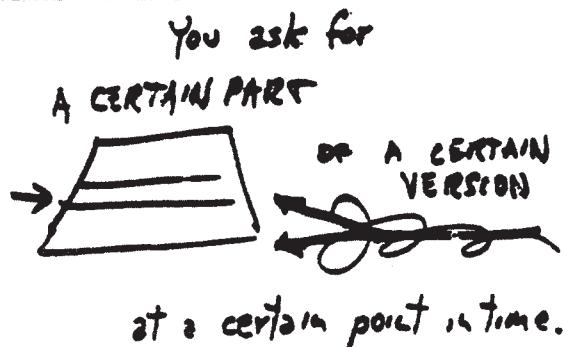
We call this *pounce*. You pounce like a cat on a given thing, and it seems to be there, having been constructed while you are, as it were, in midair. Unlike things which dematerialize when you pounce on them, like cotton candy, this materializes when you pounce on it. I can think of no other example,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

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.

\*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.

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.

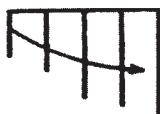
This is the *true* structure of text, because text is



#### Another Visualization

The canonical documents in this system can store the 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 time as well as space, watching the changes in a given passage as the system enacts its successive modifications.

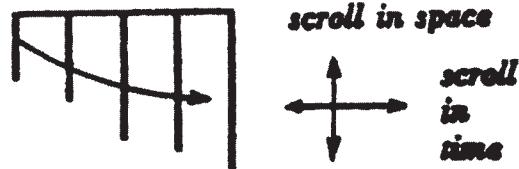


This method stores the document canonically as a system of evolving and alternative versions, instantly constructed as needed from the stored fragments, pointers and lists of our unusual data structure. Thus there is no "main" version of a thing, just the ongoing accumulation of



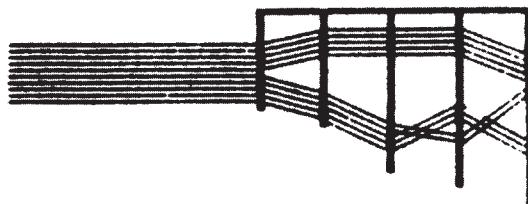
best viewed as an evolving, Protean structure.

Such storage permits easy reconstruction of previous states for mental clarification, fresh starts, and transfusions of previous ideas. It also



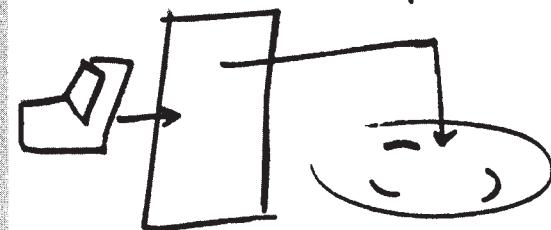
Versions of a document set apart for other reasons—"alternative" versions—may likewise be flipped through or efficiently compared side by side.

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).



449

REQUIRES TRACK-AND-SECTOR CONTROL



pieces and changes, some of whose permutations have names and special linkages. In other words, our system treats all versions of a document 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.

permits multiple uses of the same materials for alternative versions and "boilerplate."

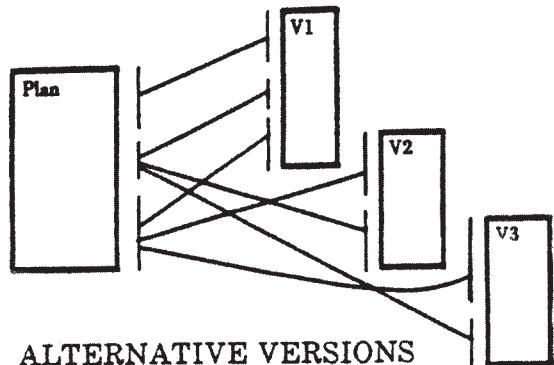
This same approach—storage as an evolving structure with backtrack—may be extended to all

forms of data that are created by individuals.

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

### 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.\*



450

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.

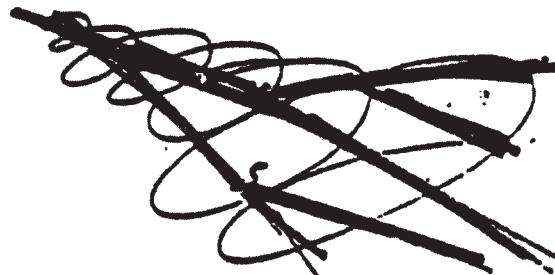
This is in contradistinction to conventional operating systems, from CP/M to OS/370, which typically deliver the whole file on every request. The illustration depicts the operating system as Frankenstein's monster with a silver tray. (These features may usually be defeated, but that's another matter.)

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

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

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

### ~~TREE OF ALTERNATIVE VERSIONS~~

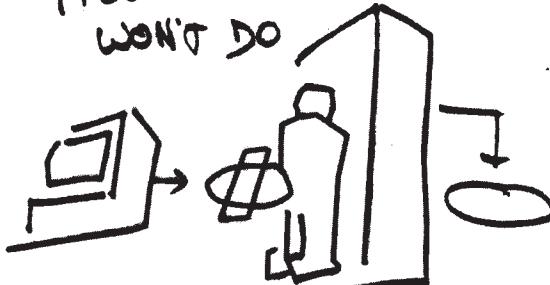


### Sameness and Difference Display

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).

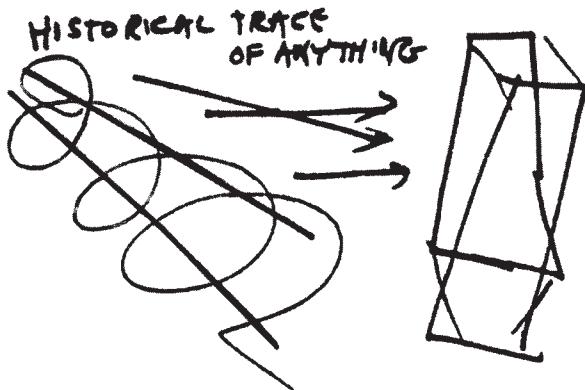
### ~~FILEWISE OP SYSTEMS WON'T DO~~



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

### 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 you 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.)

Using the data structures 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-talk 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 variant 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 non-sequential 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 these 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 *have* 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.

#### 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.

4

##### 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 *bookmarks*—places he may want to re-enter text returning to it.

##### Jump-Link

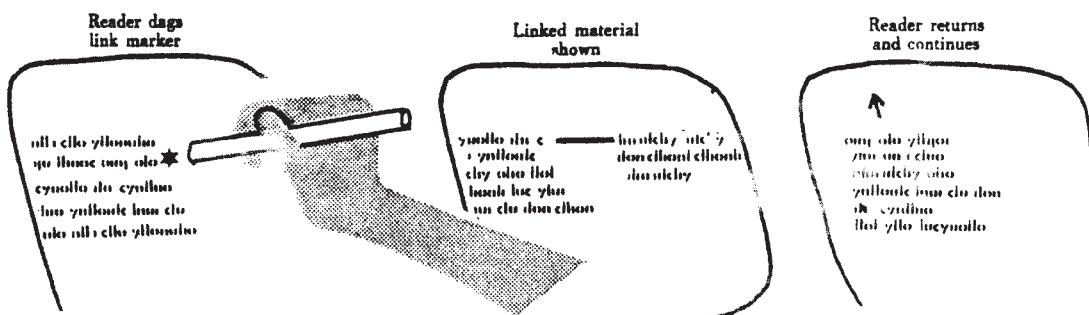
As another simple first example, let us simply think of a link as some sort of a jump opportunity, like a conventional footnote. An asterisk, say, signals that "there's something to jump to from here." If you point at it with your lightpen (or mouse or whatever), Bingo!—you're now at the footnote, or 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.

##### 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.)

452

##### JUMPING ON A LINK



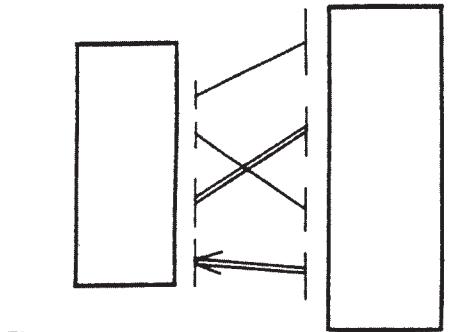
##### \*LINKS + PRISMATICS → USABILITY

Most computer schemes for linkage face the terrible problem of "updating" the links as text is modified and successive versions come

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.\*

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.

##### PARALLEL LINKED TEXT



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

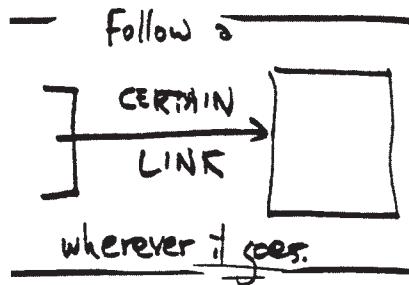
##### Hypertext

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

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.



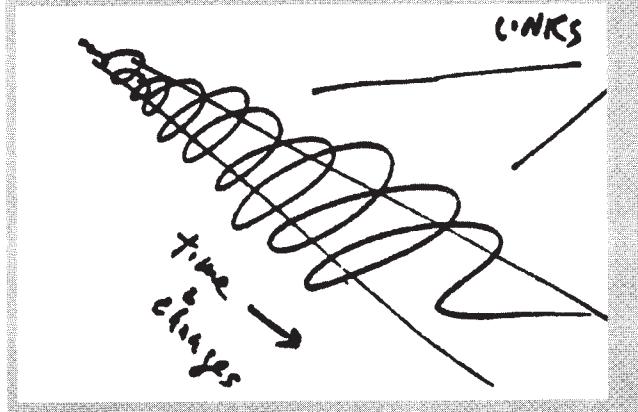
### Any Types of Links

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

In principle we allow any type of link to be defined by the sophisticated user. These include point-to-point links, point-

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

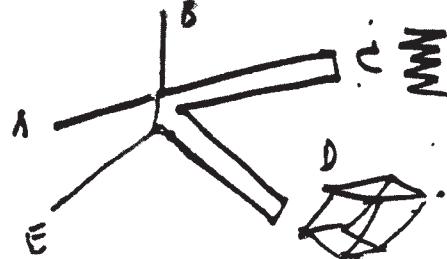
Essentially, the link seizes a point or span (or any other structure) in the Prismatic Document and holds to it. Links may be refractively followed from a point or span in one version to corresponding places in any other version. Thus a link to one version of a Prismatic Document is a link to all versions.



And any types of links may be created.

to-span, and span-to-span, having any separate names and functions desired. We also allow links with multiple endpoints.

### ANY KIND OF LINK



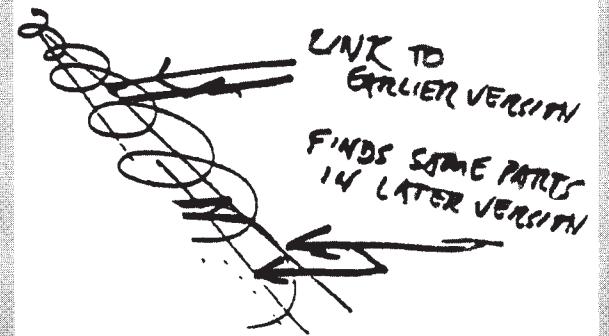
### End-Sets

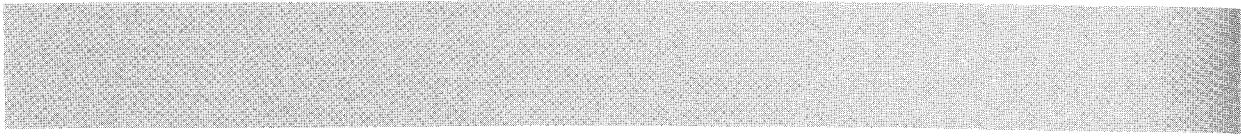
Links may attach to other links. Thus we have the concept of an *end-set*, the several types of object that a given link may attach to.

Consider, for example, an arbitrary type of link which we may call a "wuffle." A wuffle, let us say, connects a span of

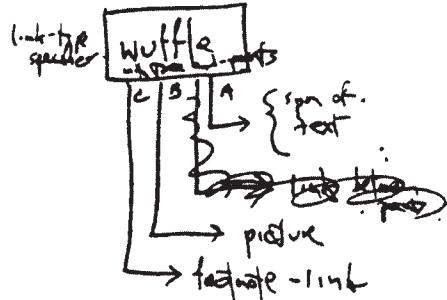
The effects, then, of links, alts and backtrack are in some sense multiplicative: together they give you a united facility of great power.

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

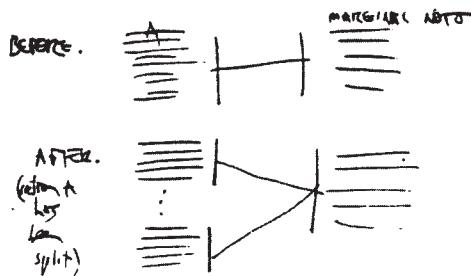




text, a picture, and a footnote. These are the *endparts*; together they constitute a wuffle's *end-set*.



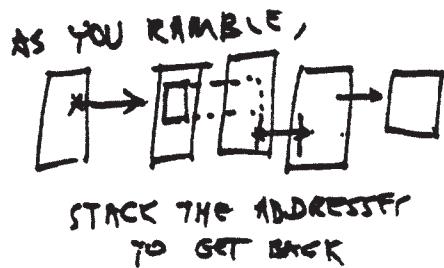
Directionality, if any, is given in the link-type definition. Note that end-parts may not hang together as they evolve (e.g. text sections):



454

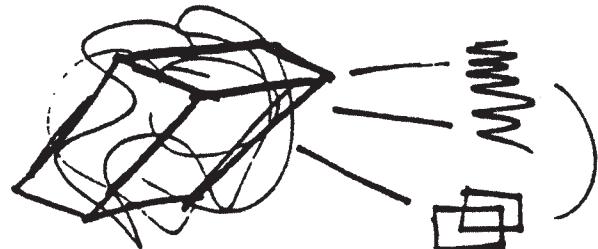
### Links and Front Ends

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.

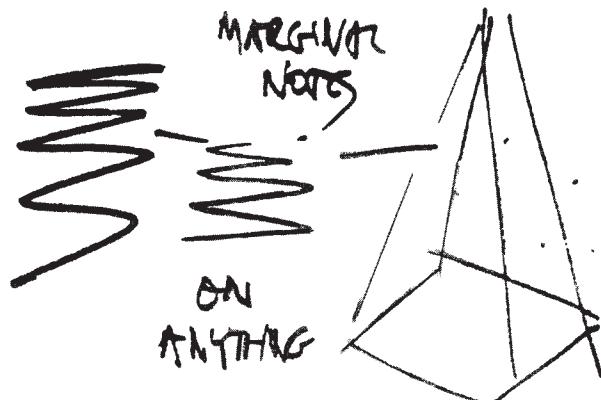


### Linking Amongst All Data Types

**ALL DATA STRUCTURES  
MAY BE MARRIED**



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.



## 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.

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.

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.

## 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.

### 6

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

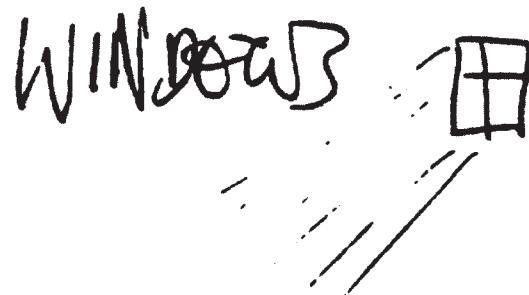
Why not allow anyone to create links *between* documents, allowing you to jump straight from one to another?

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. The integrity of this document is maintained; no one may change it but the owner.

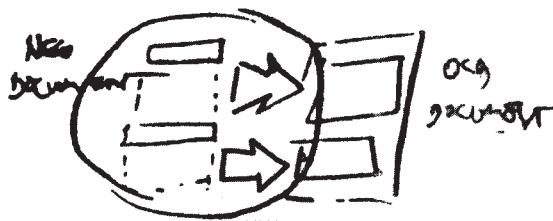
But someone else may create a document which quotes it as much as desired. This mechanism we call the *quote-window* or *quote-link*. Through a “window” in the new document we see a portion of the old.

455



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

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

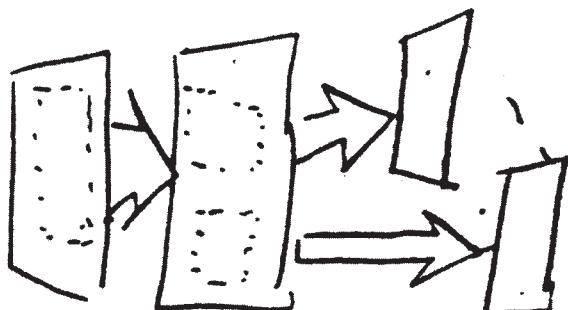


Through the same document conventions, the compound structures mentioned earlier maintain the same conventions of integrity and ownership.

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.

456



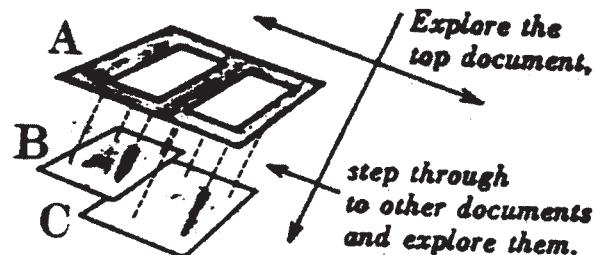
Anything stored by one user on the system may be quoted—adopted into a document—by another person writing on the system (provided the second user has legitimate access). This freedom of windowing applies, of

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

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 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.

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.

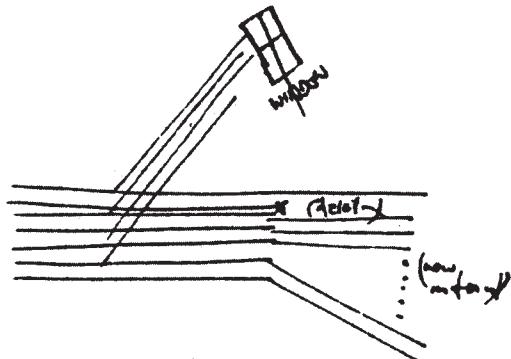
The windows of a windowing document are themselves actually particular links between documents. No copy is made of the quoted material; rather, a quote-link symbol (or its essential equivalent) is placed in the stored symbol-train of the quoting document, since no copy is made. Nor does it affect the ownership.

(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.

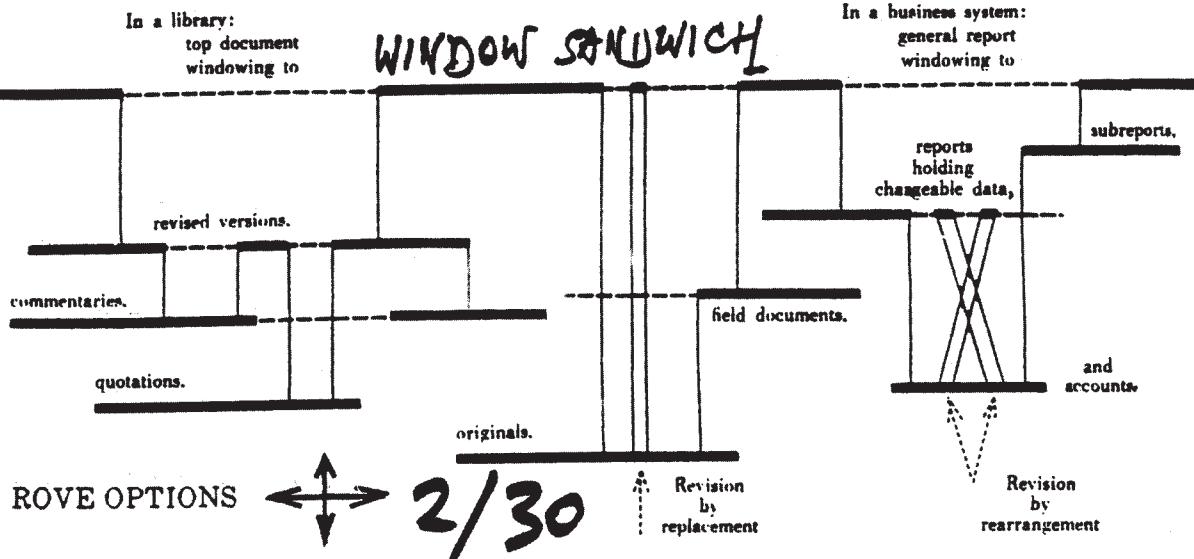


Note also, however, that a window may be fixed to a document at a certain point in time, in which case revisions

457

### LAYERS OF WINDOWING TEXT.

Each horizontal line is a document.



are seen by the user only when he asks, "What has this passage become?"

### Derivative Documents

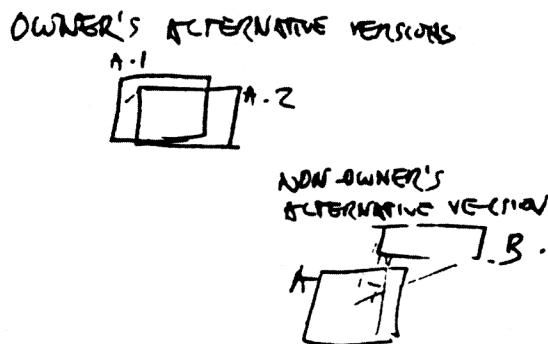
The integrity of each document is maintained by these separations: derivative documents are permanently defined in terms of the originals and the changes. (And stored on that basis.)

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.\*

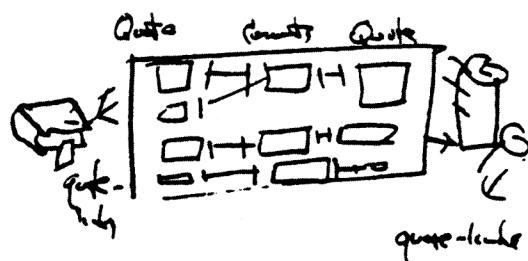
Another user, however, is free to create his own alternative version of the document he does not own. This, then, becomes a *windowing* document using the same materials.



458

### Intercomparison Documents

A document that points out relations between other documents we may call an "intercomparison document."



\*The official naming-mechanism of the system has "document" and "version" fields.

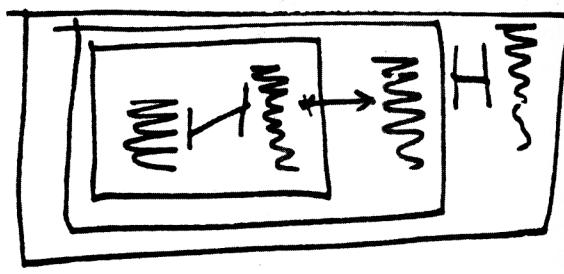
Such documents may be easily created, say, to point out relations between the Bible and the Dead Sea Scrolls.

### Compounding of Other Link Types

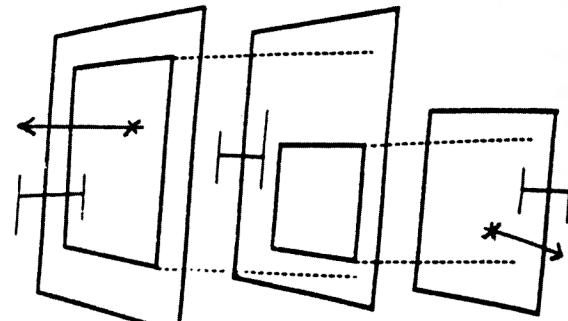
Any other link type (beside windows) may likewise go from one document to another, and interweave with quote-links.

Note that *links*, like text and pictures, *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.



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



COMPOUND NESTED LINKS

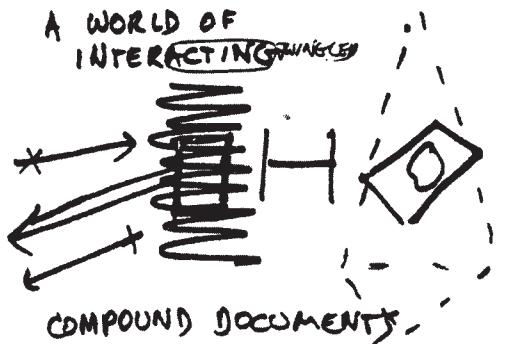
### 7 Electronic Publishing:

#### Making the Literary System Universal

Beyond its use as a private facility, we intend that this system be usable as a publication system.

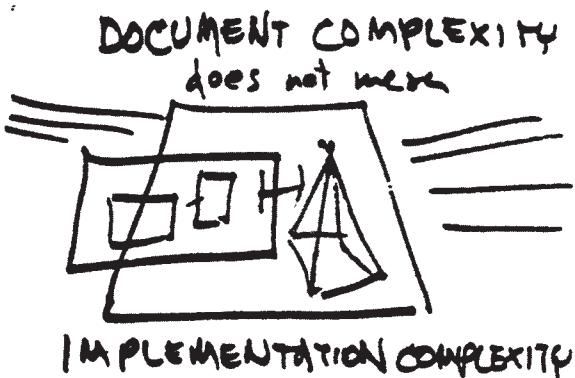
#### An Interesting World

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.



#### ... But a Simple One

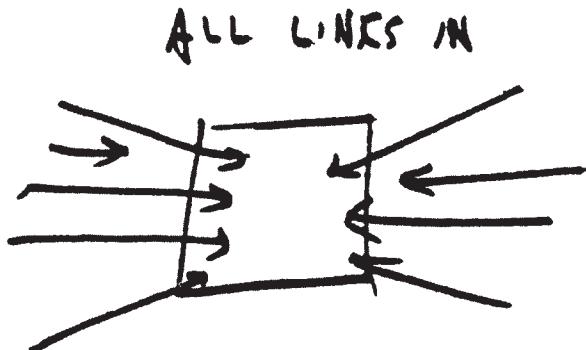
This world nevertheless remains simple in design. The virtuality is simple in structure and repeats in layers.



### 7

#### 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 are once, followed by the linked documents themselves if desired.

This must all be fast enough to please the impatient online 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.\*\*

#### Directories and Categories

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

\*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.

Thus a carefully designed system of publication, surprisingly like that of paper, has been worked out.

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.

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

Private documents can link and window to public ones.

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 *out of the system level*. 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.

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.

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.

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.

### 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 Philips. We are often asked whether these "videodiscs" will be useful for our system, and the answer is yes, but they aren't videodiscs.))

## 8 Distribution and Networking

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

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

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

## 9 Vital Issues

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

Numerous issues of personal freedom are conspicuously present.

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.

### 9

#### 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.

#### 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.

#### 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.

