INFORMATION PROCESSING 80, S.H. Lavington (ed.)
North-Holland Publishing Company
IFIP, 1980

Invited Paper

REPLACING THE PRINTED WORD: A COMPLETE LITERARY SYSTEM

Theodor H. NELSON
Project Xanadu, Box 128, Swarthmore, PA 19081, USA

Developments in computer technology now make it possible to store stupendous amounts of written material economically. This leads to radical visions of what may come to be. However, if paper is to be supplanted, it has got to be by something rather like the paper systems we now have, even if in some abstracted and unfamiliar form. What, then, have we? And how abstract and extend it? Our model is "literature"-- defined in a somewhat unusual way.

Under guiding ideas which are not technical but literary, we are implementing a system for storage and retrieval of linked and windowing text. The "document," our fundamental unit, may have windows to any other documents. The evolving corpus is continually expandable without fundamental change. New links and windows may continually add new access pathways to older material. Fast proprietary algorithms render the extreme data fragmentation tolerable in the planned back-end service facility.

THE POSSIBILITIES AND THE PRECONCEPTIONS

Vast text storage, and communication techniques for reaching into it, are a new technological development to be coped with-- which very few people were expecting.

So-called "word processing" systems have already greatly changed the handling of the written word in the office environment; now the question is how to extend this approach. It now is feasible to replace large-scale systems of in-house publishing with computer storage (Lancaster, 1978); and it will soon be practical to do the same for publishing to the open public.

Indeed, true electronic publishing is feasible now-- that is, public-access document systems with digital storage and demand service to the open public.

The problem is not electronic. It is not "software," meaning procedural obstacles to implementation. The problem is conceptual. If such systems are to be promulgated to a wider public-- no longer just in-house-- they must be clear and simple to use, yet offer powerful new features. They may not merely be a clumsy imitation of paper systems.

This is a design problem. It is a problem of creating a conceptual structure, an organizing system of ideas and methods and patterns of work.

There is no universal or obvious approach to this problem, though numerous candidate approaches exist. Various parochial disciplines and ideologies within the computer field, and other fields, have styles of thinking that seem to speak to this problem. How clearly they speak to it is a matter for careful thought.

In word processing, though screen methods for actual edit have become streamlined, all systems appear to bog down at the borders of a document. The conventions of computer storage are little improved, and so secretaries must deal with all the usual file conventions, problems of file naming (and keeping track of names), backup, and keeping backup copies safe and segregated, and so on. Few linkage facilities exist either within or between documents.

A number of on-line communities exist, particularly around such time-sharing systems as the ARPAnet. But they, too, suffer from the conventional problems of file naming and backup, in addition to the greater complexity of using the systems at all. Curiously, the users of such systems still must in most cases give their documents short names, keep paper lists of these names, and juggle backup copies in the same way that secretaries do.

Systems for "electronic mail," though much publicized, are still largely in the telegraphic tradition.

The "office of the future," we are told, is quite near. There seems to be little agreement on what it will be, however. Some say it will have optical scanning for input and paper for output. Some say it will have an "information manager" who will try to keep it all working and keep the system brought up to new revised specifications. All the time. Some say it will not even happen, whatever it is.

Some researchers, impressed by the work of Douglas C. Engelbart and his "NLS" system at Stanford Research Institute, have proceeded on the assumption that good tools for the intellect can be built with good text systems. The NLS system is essentially a community of shared files, with facilities for rapid search and linkage. As implemented it requires extensive training; but Engelbart's ideas have been widely

influential, and would seem to lead toward simpler systems for wider user groups (2,3,4).

A recent development has been the teleconferencing system, most conspicuously developed by Murray Turoff at the New Jersey Institute of Technology. As most often seen, the teleconferencing system is a setup for accumulating messages sequentially from different participants. These messages, always added at the end of the accumulating scroll of text, tend to take on a character not unlike informal conversation. In some versions, however, they have become complicated by tricky sets of rules and an elaborate supervisory function for those people designated as conference leaders.

The field of "information retrieval" appears to have stabilized into a certain basic form. This is usually the storage of a number of items of text which can be scanned on the basis of a formalized user query. (A commercially successful example is the New York Times data bank.) Usually what are stored are article summaries, but this approach could in principle be extended to fulltext as well. However, the training to use the search and query methods of such systems tends to be long and arduous, the cost is high, and users express disgruntlement about the unpredictability of results.

Advocates of artificial intelligence have repeatedly informed us that "everything we want" will be forthcoming in an unspecified manner at an unspecified date in the very near future. However, we are assured that it will involve interactve dialogue with some kind of intelligence whose internal workings and system of thought need not concern us. This Softbeing will understand us perfectly, foresee our needs, wish to keep us completely satisfied, and have nothing to hide.

In summary, I would say that the situation is one of total confusion and odd preoccupations. I see no obvious, let alone conceivable, way that all these concerns and obsessions can be comprised into a single outlook, let alone a common system. (Nevertheless, there are respected researchers, such as J.C.R. Licklider, who seem to think they can (5).)

Perhaps the apparent complication and mutual intractability of these different approaches is related to something else: they all have rather little relation to our present uses of paper.

INTERACTIVE SYSTEMS AND THE DESIGN OF VIRTUALITY

Our approach to a computer design we call "the design of virtuality." By virtuality we mean the seeming of an object or system, its conceptual structure, its atmospherics and its feel.

Every object has a virtuality, a seeming. Natural objects are more or less what they seem to be; man-made objects are not. The virtuality of a house, or an automobile, is what the designer made it-- the structure and qualities that were chosen, and the techniques by which they were realized.

The closest analogy to the design of interactive computer systems, I think, is the making of movies. What counts is effects, not techniques. We are not concerned with just how a certain effect is to be achieved, so much as with what effect is wanted.

An effect is something intended to take place in the mind. Suppose the movie effect desired is a sense of a monster approaching. This can be done by showing a man in a lizard suit-- yaargh-- or animated puppets, or by showing the fright of a person who sees the monster. In other words, a variety of techniques may be selected toward a common effect.

The design of an interactive computer environment, similarly, should not be based on particular hardware, or a particular display device, or a programming technique. It should be based on the intended effect in the mind and heart of the viewer. ("Heart" here is added because we are too seldom mindful of the emotional component in a user's reaction.)

Another way of saying this is that the "systems analysis" for an interactive system should deal with the mental space of the user's experience.

The process is a cycle: study, and design. First we must study the approximate structure of whatever we are designing, and roughly what it is about. Then we design, that is, look to see how the computer's capacities may be made to assume a similar conceptual shape. We will return to the system design process, and its dialectical character, later in this article.

There is one other key constraint in system design: conceptual simplicity. If any but highly-trained people are to use a system, it must be extremely simple. It must be simpler by far than anything computer people are accustomed to designing-- a factor of ten, let us say, simpler than what a computer hacker considers "simple."

Popular lore in the computer field holds that simple systems are not "powerful"-- where powerful seems to mean "allowing concise macro-language programming." (This is evidently the view of those who consider TECO a powerful text editor, or, indeed, a text editor.) We believe that true power, meaning easy and focussed control by the user on what he means to do, is not merely compatible with simplicity, it requires it.

This has been a preface to understanding our design. Our approach to electronic text has been to look for the hidden nature of writing as a whole, and the way it is used, to find a paradigm for this design. The virtuality we have designed reflects this endeavor. (In point

of historical fact, this philosophy of virtuality and simplicity arose in parallel with the development of the design, but we think we now know how to design any other system on the same basis.)

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 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 a 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.

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 book and refers to it. 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 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 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 of Project Xanadu 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.

But as we said earlier, the design of virtuality is a dialectic. First we must study what there is; then we must venture a design in response to what is. And that design is not dictated by what we have seen, any more than the design of a house is dictated by a hillside. There are only hints.

THE SYSTEM DESIGN

By the "design" of the system we mean its conceptual architecture: the basic ideas of it. But with many systems the general system of concepts is filtered through innumerable complications and accidental features. In this system the basic ideas are the only ideas.

Our approach, then, is to consider all documents as interconnected, or potentially interconnected. Not just documents, but linkages as well, are the fundamental elements of the system. (Links are actually parts of documents, as will be seen later.) It is the different types of links that give both power and complication to the system.

Our system, the Xanadu* Hypertext System, is intended to store a body of writings as an interconnected whole, with linkages, and to provide instantaneous access to any writings within that body.

For this system we have formulated a basic linkage structure that we think meets all the needs of literature as we understand it, including business literature.

Beyond this general idea there is little further to expound, except the link structure and the document conventions, which are all the user really has to learn. Since the network is essentially invisible, all we have to discuss are these links and documents.

LINKS AND COMPOUND OBJECTS

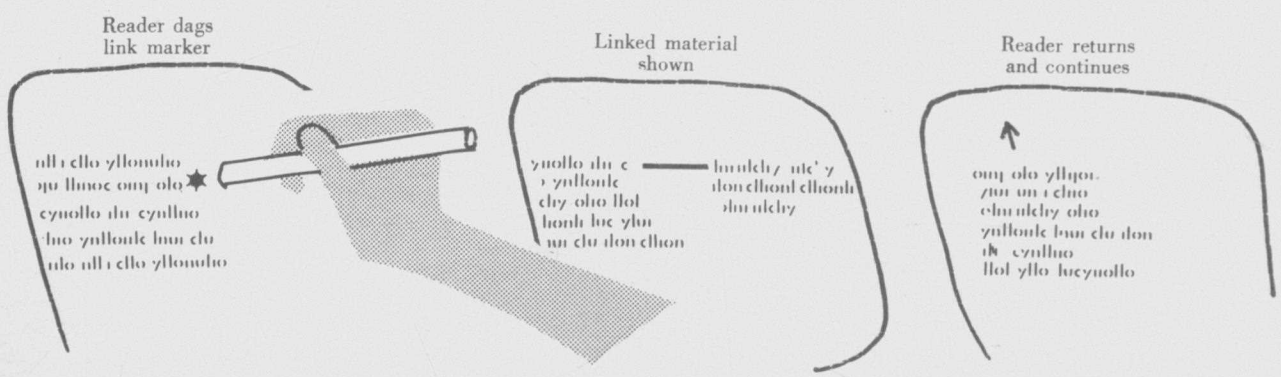
A link is simply a connection between parts of text. It is put in by a human.

A user, on contemplating any two pieces of text, 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.

---

* "Xanadu" is a trade and service mark for computer products and services offered by Project Xanadu, Swarthmore, PA.

Figure 1. JUMPING ON A LINK



PARALLEL LINKED TEXT

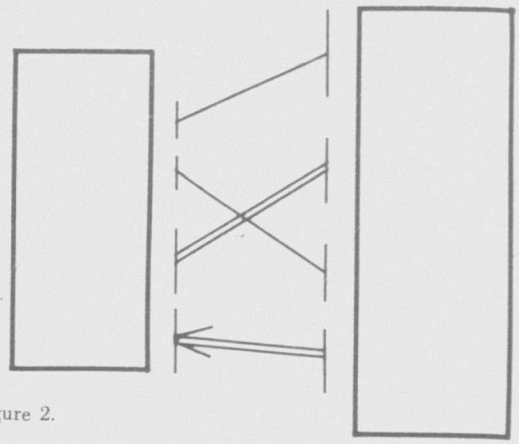
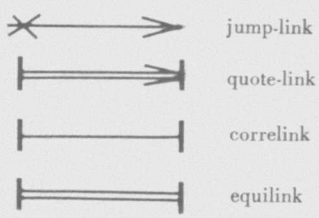


Figure 2.

THE SPECIAL LINKS

This is essentially the basic system. However, its extensions and ramifications, and the unfolding link-types and their capabilities, hold many non-obvious challenges. Certain link types, which we call the "literary set," are the main ones. This is because they have direct analogies in existing literature as everyone knows it.

Figure 3. THE LITERARY LINKS



They are (1) the jump-link, corresponding to the footnote; (2) the quote-link, corresponding to the printed quotation; and (3) the correlink, resembling the marginal note, or "corresponding part." A fourth type, the equilink, will be discussed a little later.

The jump-link is essentially a pointer to a related item, something like a footnote. A reader, after satisfying himself by reading the related item, may push a "return" button to return to the main text.

The quote-link makes it possible for a piece of text to be in two places at one time, but with the document of origin still master of the quoted material. (Quote-linkss are automatically maintained by the system, and automatically preserved through a variety of editorial changes, even those involving fragmentation.)

The correlink places a segment of one document "next to" a segment of another document. (Unlike quote-links, correlinks are put in by the user, and in a sense resemble index-tabs or highlighting, being markers to accentuate structural relationships. While correlinks survive fragmentation, it is with some difficulties.)

Correlinks may be used for annotations, or to mark correspondences of any kind. They may also be used to connect parts which correspond to other parts or wholes, as a title corresponds to its document.

COMPOUND DOCUMENT-CLUSTER

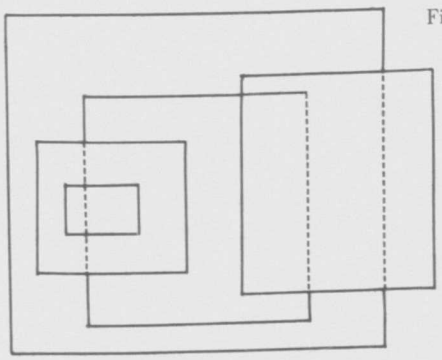


Figure 4.

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.

(Many other types of link are allowed within the system. 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. However, the functions of the literary links are utterly unique, and we wish to focus on them here.)

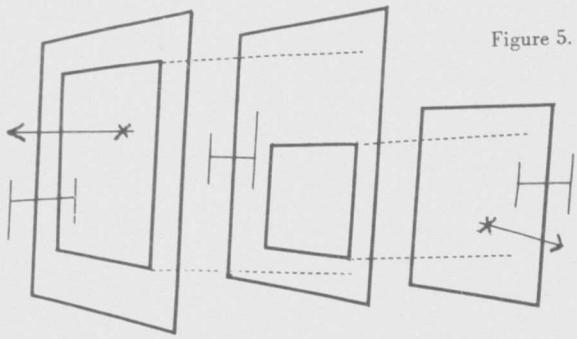


Figure 5.

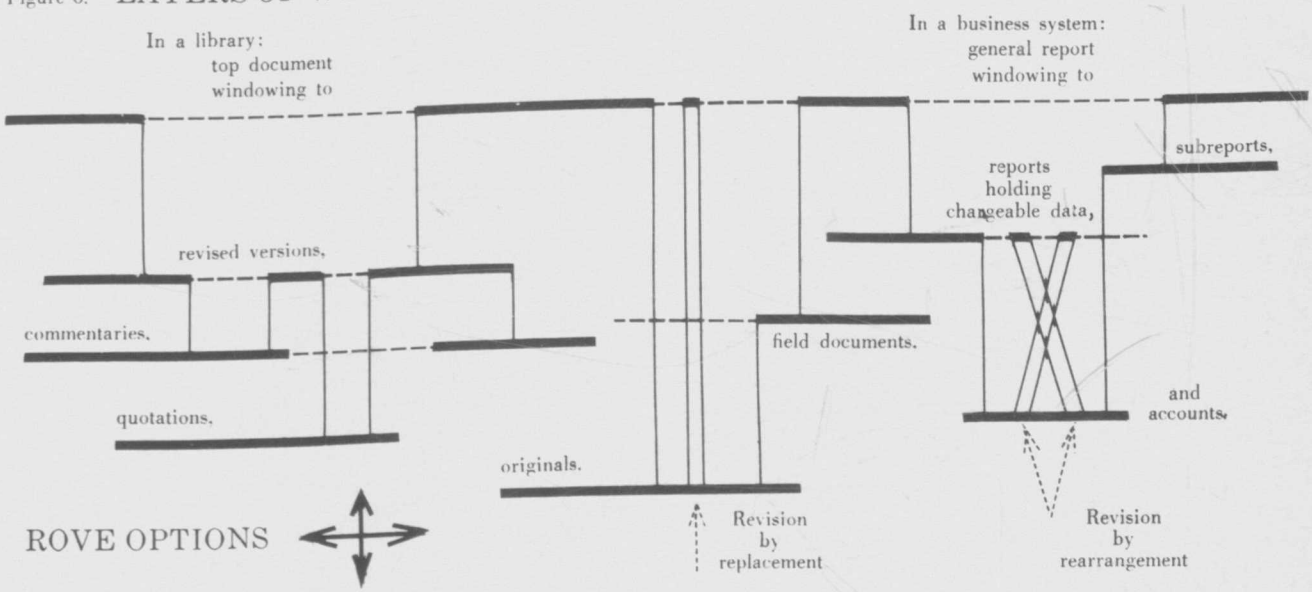
COMPOUND NESTED LINKS

It should be noted that quote-link and correlink are unusually privileged, in that they may applied to parts and to other links to any remove. They are also extensible to all other forms of computer-stored information: graphics, musical scores, and any other form of symbolic materials, whose interrelationships may be shown by these links.

THE DOCUMENT CONVENTION

An interesting choice has been made in the design of this structure. We call a thing a "document" whether it contains text, or links, or both. A document is something designated by a person to be a document, containing text and/or links that he has created. The links are part of the document, though whatever else they link to may not be.

Figure 6. LAYERS OF WINDOWING TEXT. Each horizontal line is a document.



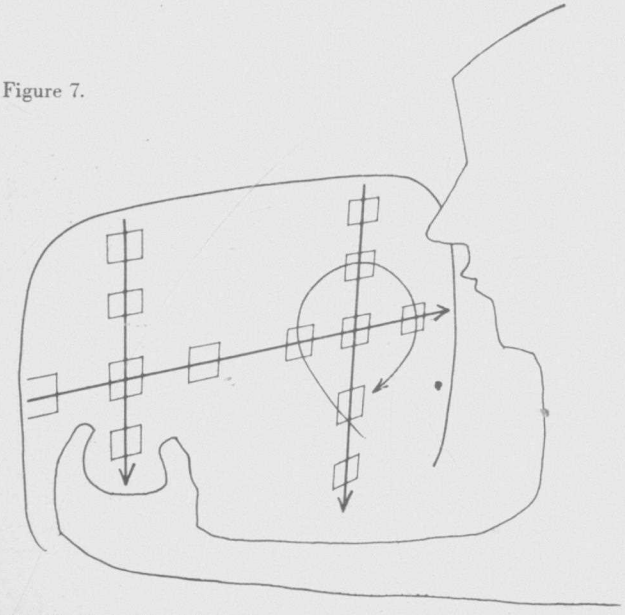
By this convention, then, everything in the system is a document and has an owner. No free-floating materials exist.

Through the same document conventions, the compound structures mentioned earlier maintain the same conventions of integrity and ownership. But one document can be built upon another, and yet another document can be built upon that one, each having links to what was already in place.

ORDERLINESS, EXTENSIBILITY AND GENERALITY

Many have admired Vannevar Bush's postulated "memex" of 1945, but most have supposed out of hand that it was impossible. Bush, it may be recalled, spoke of a hypothetical console that would hold all the documents a person ever read or wrote, and allow him to study "trails" of documents, and put in new trails himself.

Figure 7.



BUSH'S MEMEX:
documents are stored in "trails" of unspecified structure, which may be cross-threaded in additional sequences. User keeps track through code book.

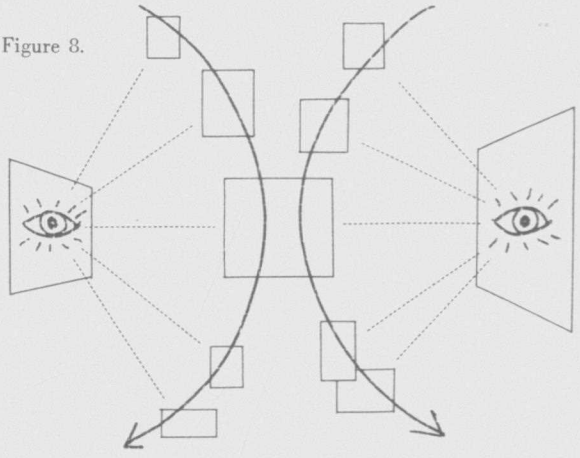
The memex was meant to be a microfilm device, but many have wondered how the basic ideas in its design could be transposed to a computer technology. This has been a large force behind "information retrieval," but that effort has turned much more to techniques of indexing. The fundamental question, however, is no longer whether such a thing could be done, but how such a thing might be structured.

Our system is an answer. Essentially, it is a full-blown memex system with an orderly set of conventions. But where Bush conceived his memex as chiefly a private desk for working with a personal microfilm collection, we see ours as a potentially universal system for both public and private use. It has been designed for indefinite expansion.

Because of this contemplated scale, the system obviously has to be extremely orderly in concept. This we think we have achieved.

The system seems to offer power for the ordering of the most complex mental problems with a minimum of complication and distraction. (At the same time, as already mentioned, it abolishes many of the trivial complexities of keeping track of evolving files.)

Figure 8.



GENERALIZED MEMEX:

trails are themselves documents.

As an example, let us consider using it as a programmer's console. Suppose that a programmer begins by with a written description of the program he intends to write. (Sometimes this is called a functional specification.) This description is entered into the system. He then creates a detailed breakdown of the parts this program is to have; this description too is entered into the system, and correlinked to the functional specification.

Let us say our programmer now begins to write code. He simply enters it side-by-side with the structural specification. This "side by side" relation is maintained by the system through the correlink mechanism. Even though the programmer may rearrange his materials to his heart's content, the "besideness" is maintained among all the appropriate pieces.

Now the programmer tries compiling part of the program, and gets object code. This too is filed, correlinked to the source code. All the work in progress is maintained automatically in this side-by-side structure, no matter what organization or alternative versions are created. Even error messages engendered by the compiled code may be automatically correlinked.

Each of these separate correlinked pieces may be thought of as a "column" of text, whose portions are "next to" their relevant companions in other columns, though their sequences can be different.

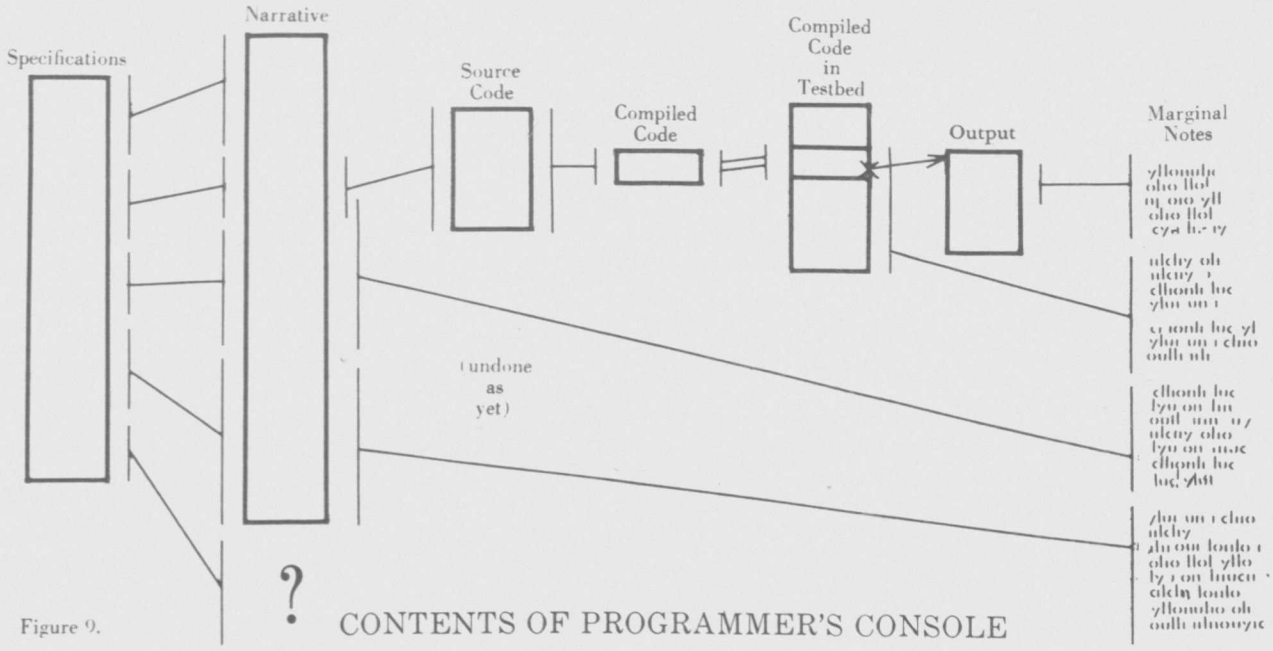


Figure 9. CONTENTS OF PROGRAMMER'S CONSOLE

Note that if the programmer chooses to write a program two different ways, effectively creating alternative versions of it, the different versions may neatly be shown on his display device and their appropriate parts referred to and compared as he may wish.

While we need not go indefinitely into this example, it should be clear that this design furnishes a flexible container for reading back and forth between the relevant levels and structures of a typical complex activity.

COPIES AND BACKUPS, HISTORICALS AND ALTERNATIVES

Our system does not merely handle a document as text and links. Rather, it 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 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.

The alternative versions employ one other link-type, not discussed earlier, which our data structure uses for alternative and historical versions.

The correspondence between the same thing in two versions of a document is kept track of automatically. An abstract link stretches between these two parts.

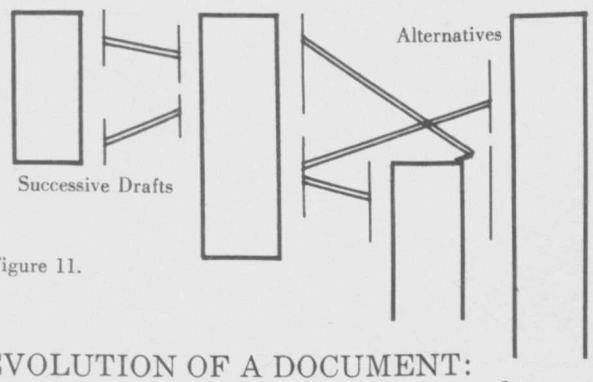


Figure 11.

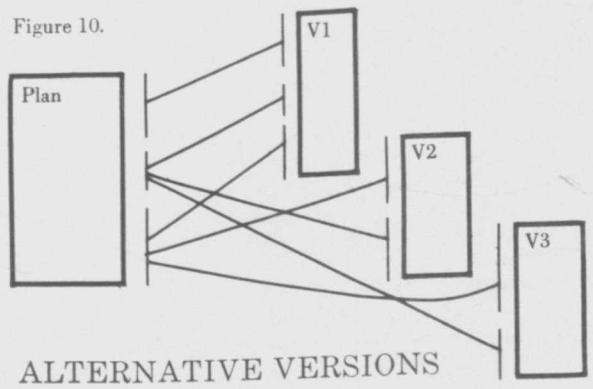


Figure 10. ALTERNATIVE VERSIONS

EVOLUTION OF A DOCUMENT: equilink persists across changes and alternative versions

This link between the same thing in two versions of the same document we call the equilink. Our system automatically keeps track of equilinks when more than one version of a document is shown. The equilink differs from the quote-window in that no occurrence of the section has precedence as the original or owning document. (Conversely, the quote-link may thus be seen as a directional equilink.)

The scope and generality of this feature may not be immediately obvious. To many programmers it seems like "too much trouble" to create facilities of this kind; however, now that it exists, it may be used for any casual purpose of the user.

Anyone who has worked with computers is accustomed to the frequent and disagreeable problem of keeping backup files. Many of us also maintain files of previous versions, and/or alternative versions of files for different purposes. These trivial problems create endless annoyance because they are not ordinarily taken care of automatically.

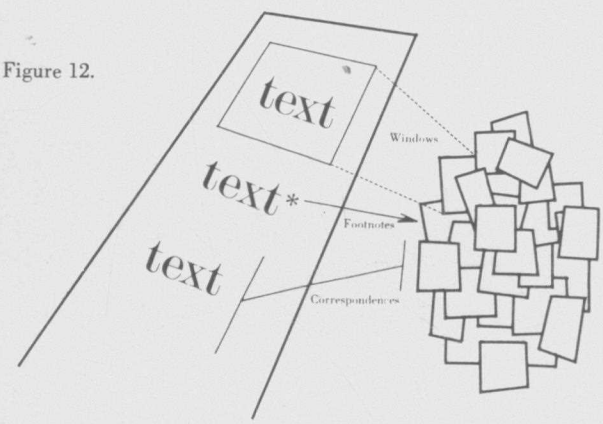
The system described would seem to cut to the center of this problem.

FROM A CONSOLE TO A NETWORK

So far 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. But given today's network technologies, this is not a real restriction.

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. 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.

Figure 12.



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

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.

MULTI-PERSON USE

For the time being we will ignore the problem of privacy, and assume that all users are freely sharing their work.

Anything stored by one user on the system may be quoted -- adopted into a document-- by another person writing on the system. No copy is made of the quoted material; rather, a quote-link symbol (or its essential equivalent) is placed in 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: in our planned service, a standard proportional copyright fee is paid automatically by the user every time a fragment is summoned.

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

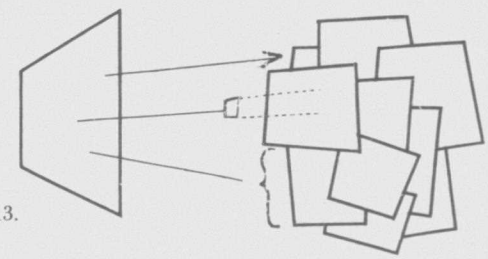


Figure 13.

ANY SUBSET, SUBSTRUCTURE OR GROUPING

PHILOSOPHY OF THE PERPLEX

Often the truth about a subject is difficult or impossible to find, though a great deal of information about it is on hand. Frederick C. Crews has implicitly proposed the term "perplex" for such a body of information, in his masterful academic satire, The Pooh Perplex (11).

Intuitively, we ought to be able to use computers to help us sort out and order the complexities of what is written, so that our grasp of it becomes firmer and clearer. I have proposed the term "thinkertoy" for such a facility (10); more recently such terms as "decision support system" have appeared in the literature. But what has been less clear is the nature of what such systems should be like.

Our system facilitates multiple interpretations of the same material. Whatever is stored may be seen as a compound object, either organized in different versions or viewed through other documents. Each may independently represent a different point of view. Thus users may highlight different interpretations of the same material, by quoting or linking in from different documents. Thus it seems to be a genuine thinkertoy for strong and subtle intercomparisons.

Figure 14. TWO VIEWS OF A SUBJECT.

We spoke earlier of the unending change of ideas, the way in which a given field is constantly subject to reinterpretation. It is a tradition of western thought that such reinterpretation is always possible, always going on. But how can a literature that has been described in one way be reinterpreted in another, without total rewriting?

Within our system, the user may make marginal notes and new documents that ease his task in totally reinterpreting (or "newly interpreting") the material before him. He has only to make his own summary of another piece of writing-- and indicate the pathway by a correlink. A reader may then summon the corresponding part at any time he wishes to confirm the interpretation.

The improved visualization and control of alternative theories and viewpoints, of disparate and corresponding ideas, should give a person a broader grasp of all everything he reads and thinks about. By providing such tools ready to hand, we think we are contributing to a new way of seeing.

A RADICAL IMPLEMENTATION SEQUENCE

The system as it is currently being implemented is based upon this structure.

Since conventional operating systems work with conventional files, new approaches had to be found.

Now, the normal way to create a new program in the computer field is to implement it on an existing computer setup. A computer setup consists of a computer and its main control program, or operating system, which handles all file access and update. However, since our approach required a radical redefinition of file storage and operations upon what the files contain, the focus of the enterprise has been on the redesign and re-implementation of what most computer people have thought was completely settled, the file system. This in turn requires the creation of a whole new operating system, since the operating systems presently available rest upon a conventional view and implementation of file structures. These have been non-trivial obstacles.

The current version is now in code in a suitable systems language, and we expect to demonstrate the major functions of the system in a reasonable time.

PUBLICATION THROUGH THE SYSTEM ON THE POSTULATED NETWORK

The system's design is a unified whole, but we may think of it as the basic conceptual structure, plus a technical structure which makes it possible, and a contractual structure which making 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 larged 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.

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

Any user may store anything on the system. Unless specified otherwise it is a private document. If the user chooses to publish it, however, 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. The author signs an "I hereby publish" form, after which not only is the document universally available, but its author may not withdraw it except by lengthy due process. (He may readily publish a superseding document, but the former version remains on the network.)

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.

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 documents, private or not.

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 other of a published work is relinquishing the right to control links into that work. This relinquishment is part of the publishing contract.

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.

We will recommend certain programs for use on the user's terminal or personal computer, but these may be created by any party. Approved terminal programs may be offered certain trademark advantages, but no terminal behavior can or will be forbidden.

NONCOERCIVE IN USE

We see this system as offering remarkable power to all users with the greatest possible freedom of use.

THE SYSTEM'S FUTURE AND SWEEP

We have created this system intending to offer a viable alternative to all forms of reading, writing, archiving and study now handled by methods of paper. Through the system it is possible to mimic, perhaps viably, many aspects of the great society of paper: books, magazines, private notes, copyright, royalty, archiving, and roles for author, publisher and critic.

We want it available to everyone at $2 an hour, and to assure freedom of speech on the system, the integrity of copyright, and other high-minded objectives with respect to its broad future use.

Obviously this is a somewhat ambitious plan, and we cannot judge its viability ourselves. But even given more moderate goals, it seems to be a versatile structure for other purposes. For instance, it would seem to be a good public-access memory service, offering a back end with much greater flexibility than standard storage. Objects can be stored "seamlessly," and the user need have no concern for their size, naming of alternative versions, linkage and the like. Thus it would seem also to be a favorable storage structure for naive-user systems of all kinds, by its removal of various levels of complication.

It should go without saying that the system may be used for all other forms of linked data base, including animated graphics, which we hope will be an important component of future educational and leisure systems.

Such an approach offers to standardize, not languages or processors or algorithms, but the storage form of linkable and complex materials, and terminal interfaces for their exploration. This would seem also to be a worthy goal.

It may be noted that the system adapts readily for purposes of "electronic mail," using the null adaptation.

CONCLUSION

Text is the self-portrait of human thought; more precisely, it is the ordered presentation of the results of that thought. Specific textual conventions have evolved in different aspects of human endeavor, but to study any of them by itself is misleading, like studying only one part of the body or only one sex of a species.

The computer field has gradually become aware of text problems, but most computer people see them as independent areas, like "word processing" and "electronic mail" and "information retrieval."

And so it is that computer people have for the most part never looked at the whole picture. Conventional system designers have approached small subsets of the grand text problem, and the resulting designs have tended to be complicated and cumbersome. Often they require, not only a tangle of specialized programs and user languages, but new social roles for supervisors and service personnel, since ordinary people cannot be taught their use. System complexity appears to rise in proportion to system size, or worse, by some power of system size. But a little thought shows at once that this cannot be permitted. At that rate nothing is going to work.

The view I am suggesting is that the problem is unique, singular and enormous, and the solution can therefore only be unique, singular and enormously simple. I think there are not many text problems but one problem, the text problem, which is the grand interplay of written materials, their interconnections, and the minds that play on these interconnections like harp-strings.

There is a specialty in the computer trade called "system design." Now, we all design systems, but is there a right way? (In some cases a system is intended to do something utterly new, and so there is nothing to be studied or replaced, but that is not germane to our problem.)

By some accounts, system design is the study of existing methods in some area of human endeavor, and the translation of these methods to a computer equivalent of some kind. But this broad description is not very helpful.

To observe and copy is not enough. True, the job of the system designer is in part to observe what people do in an existing system and take note of all the different activities that he sees-- no matter what the people think they are doing or seem to be doing. But the job of the actual design requires more. A designer necessarily makes compressions and adaptations. This is the creative part. And the designer should seek simplicity.

There is no reason, in systems design, to ratify and perpetuate those individual and local complications of life which have arisen in various contexts. Just as the scientist seeks generalizations, unifying ideas which summarize and compress all the varying details he may observe; so the system designer seeks also an underlying structure in whatever existing system he is adapting to the computer. But this structure is not merely empirically found; in part it is imaginatively created, and represents the designer's conceptual encapsulation of what he thinks is going on, and what he thinks should go on.

I suggest, then, that it is the designer's task to find simple structure wherever it may really lurk, and even to create it when he can, though it wasn't there before. This is especially true where non-technical people are going to use the result. Because only truly simple arrangements can be extended, and transposed, and elaborated, by ordinary people in the thoughtful pursuit of their concerns.

Only the simplest of conceptual structures can be extended across all the known functions of text. But we can have simplicity along with power and subtlety; and from extensible kernel structures provide all the functions now available in text systems, and many that are not available but would be more than welcomed. Those are the premises of our design.

This system has been under independent development for twenty years. Its marked departure from other systems and approaches has been greeted with indifference, disbelief or mystification. The only similar system known is Vannevar Bush's hypothetical "memex" system, postulated in 1945, of which we believe this to be the only existing instance. We hope no other will be needed.

While many other features are included in the system as designed for public operation, the central idea is that this conceptual structure (or virtuality) is extensible across the entire range of uses already known for the printed word. What its social effects will be cannot be foretold.

We would like this to be a principal publishing means of the future.

REFERENCES

1. Vannevar Bush, "As We May Think." Atlantic Monthly, July 1945, 101-8.

2. Douglas C. Engelbart, "Toward Integrated, Evolutionary Office Automation Systems." Proceedings of the October 1978 Joint Engineering Management Conference, pp. 63-68.

3. Douglas C. Engelbart, "NLS Teleconferencing Features: The Journal, and Shared-Screen Telephoning." Proceedings of 1975 COMPCON, September 1975.

4. Douglas C. Engelbart, R.W. Watson, and J.C. Norton, "The Augmented Knowledge Workshop." Proceedings of the National Computer Conference, 1973.

5. J.C.R. Licklider, Libraries of the Future. MIT Press, 1965.

6. Frederick W. Lancaster, Toward Paperless Information Systems. Academic Press, 1978.

7. Theodor H. Nelson, "The Hypertext." Proceedings of the World Documentation Federation, 1965.

8. Theodor H. Nelson, "A File Structure for the Complex, the Changing and the Indeterminate." Proceedings of the Association for Computing Machinery, 1965.

9. Theodor H. Nelson, "A Conceptual Framework for Man-Machine Everything." Proceedings of the National Computer Conference, 1974.

10. Theodor H. Nelson, Computer Lib. Published by the author; available from The Distributors, 702 South Michigan, South Bend IN 46618.

11. Frederick C. Crews, The Pooh Perplex: A Freshman Casebook. E.P. Dutton & Co., Inc., 1963.

12. Theodor H. Nelson, Literary Machines (in preparation).

ACKNOWLEDGMENTS

I would like to thank the principal technical contributors to Project Xanadu, especially John V.E. Ridgway, Cal Daniels, William F. Barus, Stuart Greene, Mark Miller, and Roger Gregory. Special thanks are also due to Theodor Holm, Robert W. Fiddler, Jonathan Fagin, Nathaniel Kuhn, John R. Levine, Glenn Babecki and Eric Hill.