HARCOURT CONFIDENTIAL

Theodor H. Nelson

THE XANADU SYSTEM: A Discussion Paper

23 December 1966

INTRODUCTION

This paper is intended to communicate with both management and programmers. On neither side is the system's practicability assured. Everything described here is possible and, I think, desirable. Whether it is feasible, that is, possible for a reasonable price, is not yet clear. How much of the system described can be implemented on contemplated equipment is hard to tell. Matters of programming cost and time inconvenience to the user may prove to be restricting.

These are some remarks about the XANADU system, a proposed setup for handling a large number of corporate and creative functions on a computer with display screens and a rich file handling capacity. All that is said here will be tentative thought and ideas on how such a thing should work. The XANADU system is supposed to be a general-purpose, highly flexible information handling system for the entire firm, especially suited to experimenting and easy usage. This paper presents design remarks on this master program, and a ticklish job this designing is, too, since it is the executive and editorial staff at Harcourt, Brace & World who must live with it and like it.

This is hard work. It is hard, not because there is a certain exact way to do everything, but because-- to be candid-- it can be done in any number of _different_ ways, and the many options are constrained only slightly by technical bounds. We could choose to pretend that "the computer" requires a certain mode of operation, even a particular posture, of the editor who will use it; but this would be a falsehood. With certain exceptions, it will do practically any darned thing an editor might want, if we are prepared to devote, sacrifice or subordinate this or that feature of the machine to the darned thing we want done.

Very well. The system we want to put up will be sophisticated (its fancy features will appeal to the most arcane); general (by making little arrangements this way and that you can make it do service in many different guises and stances; useful, so that its features are relevant to the comparisons and vantages you need it for; and pleasant, aesthetically attractive, simple, fleet, and fun.

The basic idea is to build one system so thorough and general that every text-handling application we can now think of will operate through it. This means that its underlying structure will be all the more anomalous, and unlike any other systems in this area.

Although it is being designed around a very small machine, it is intended to be one of the most powerful text-handling facilities in the world. The general idea is this: the system is meant to be typewriter, filing cabinet, and manuscript, and simulated printed page, all at once. As such, it will not only make the editorial updating of ordinary texts much quicker, but it must also provide information facilities crafted according to particular kinds of needs.

On this unusual facility any conceivable workspace, file discipline, or information structure may be created, and modified at will; no fixed ways of working will exist, except as available options. The set of text and file commands will not be an alphanumeric language unless desired by the user. The system will thus be of use for any mind-aiding process that uses text, symbols and connections.

The basic idea is to build an information-handling system to hold, display, modify, rummage through, and rearrange text and ideas. Unlike any other system that ever has been, this one's intent is to meet the user more than halfway, scrupulously avoiding the interposing distractions of user-burdening computer languages, special characters, symbolic order codes and the like. Moreover, it will be reconfigurable for different purposes: the user may define to the system the types and interconnection types of his information, the way he wants it displayed, the actions of the system, and the way they are to be chosen. Typically he may define choicepoints as "virtual pushbuttons" on the screen, and have them replaced by real pushbuttons, for faster work, when he is sure how he wants to work.

A unique part of the design of this system will be the assumption that our information does not typically come in strings, as is supposed, but is deeply and widely interconnected. Provision for every kind of interconnection must therefore exist, and "ordinary" text will be treated as a trivial case. Thus the system will provide for the creation of annotations, spinoff versions, linked alternative versions, and general hypertexts. (Hard copy will be swiftly available from a line printer.)

Nothing will be restricted in length. Input strings, sentences, chapters, and even titles may go on and on and on. This philosophy of non-restriction extends throughout the system in various ways.

As presently conceived, this system should go far beyond any existing on-line text systems in its flexibility and power. Editorial usage should be only one possible application; the intent in the design phase is to work out a generality that can be turned to many different purposes.

With all this in mind, I ask you to consider the following setup. These features will not satisfy any use in particular; rather, each user may adjust the settings for his own application, and thus we may configure it for many particular uses.

The present design is general-purpose-- indeed, its applicability is so general that from its raw structure it might not be evident that this is to have anything to do with publishing at all, let alone its initial assignments. That is because it must be designed in primitive forms that can be tailored to any specific needs, and these of necessity need to be so general that any tie to specific applications can be established later.

The screen may be thought of as a telescope of the viewfinder of a camera, and we can zoom it to any place in a text, or text complex, that we want to look. But the importance of the device is in its ability to bring swiftly together materials that need to be put together, or seen together. Not just what is sitting on the screen at a given moment, but also the things that are lurking in the wings, standing by or momentarily displaced, constitute the user's workspace.

The system as presently conceived will allow the user to define the types of information he will deal with, and how it will lurk, move and reply to his actions. It is as though he might design his own special sheet of paper, with little doors, fold-outs and overlays.

It should not be supposed by the reader that this description is a statement of what computers "can" and "cannot" do. Rather, this is the design of the system which _may_ be implemented on a computer, and we intend to circumvent the computer's traditional modes of operation, rather than defer to them.

As presently conceived, the system would be simple to operate, easy to learn, and easily taught and adapted to any person and purpose in the company that is given it to use.

~~From the point of view of the system's actual design, however, these particular subjects and~~

Four kinds of work have provisionally been assigned to this system.

Ordinary editing. Little need be said about this; it seems likely that employing the console, considerable time can be saved in the preparation of ordinary books for printing.

Test editing; test data bank. For editing tests, both in terms of revising and rearranging test items and accessing information about the file, Xanadu will probably be more useful than any apaer filing system. The same items on different versions of the same test, statistical data on their use, could all be filed in ways permitting swift access to corresponding units. Moreover, since the Xanadu structure would permit extensive cross-indexing and categorization, it could be used for the creation and maintenance of a test item file, whose parts could be selected from and rearranged with ease according to all sorts of different criteria.

Print-and-Bind Orders. One of the more difficult managerial problems in publishing is that governing orders to the printers and bindery: relevant considerations include predicting demand for the finished books, the apparent probability of demands exceeding the print order, the storage costs of incomplete material, and finally the schedules of the printers and binderies to be patronized. ~~All these different forms of information could be maintained in~~ Information on all these different circumstances could be made available for thoughtful perusal for the decision maker; they would be arranged in a Workspace that connected each type of information to the things it bore on.

Simulation of 1500 . There is some indication that the IBM 1500 Instructional System may become an influential systme around the country in the next five years. While it may not be worth our while to obtain such a system now, it may be worth our while nonethless to try preparing materials for it. Under Xanadu, it will be possible to simulate the IBM 1500, and create materials for it, without having one.

Among the other possible applications of this system are the editing of books, the design of highly complex non-book materials for publication, the facilitation of collaborative work between author and editor, experimental typesetting and making of animated movies, ~~did~~ AND aiding management in comprehending complex banks of data. Perhaps most importantly, the system should help shorten the time required to get critical jobs to press.

From the point of view of the system's actual design, however, these particular subjects and facts are unimportant. The parts that have to be designed into the system are more like the following:

1. *Text editing*. Contained texts may be modified swiftly with such instructions as "insert," "delete," "move," "copy."

2. *User-defined file structure*. The user may define his own information types and links, and the way the machine will move and handle them. These may range from an outline form to multi-part linked units.

3. *User-defined workspaces and file* disciplines. The user may define such abstract working arrangements as push-down stacks, temporary holding spaces, "folders," indexes, and windows into his file. He may even specify overall file disciplines that the system will thereafter hold him to.

4. *User-defined screen formats*. The user may "carve" the screen into separate areas where his files and workspaces will appear.

5. *User-defined branching actions*. The user may arrange interchange between various presentations by drawing branch graphs, and specifying the conditions under which the machine is to branch; these may include throttle, pushbuttons or screen pointing. This branch-defining technique may be used either for creating work-systems or for defining hypertexts.

6. *User-specified presentation modes*. The user may have text segments or streams presented as frames, drum rolls, and streamers; and specify which are to blink, how brightly, and when.

7. Continual Modifiability. The system a person is using may be modified *ad lib*.

8. Reversible processes. Both text and modification commands may be stored in such a way that changes can be undone as far back as records are kept.

9. *Evolutionary text generation*. Alternative versions may be spun off, and linked to each other so that their corresponding parts may be compared.

10. *Complex indexing*. The system will have a complex indexing ability for either straight text complexes, permitting the creation of any number of indexes for a given body of information, from which the user may immediately jump to the information being indexed. This indexing facility permits the use of lists or texts to index one another, so that the user may proceed to a corresponding part in the other text or listing. This may be regarded as "backwards" and "forwards" indexing, but that implies a priority which may not

necessarily exist in the material.

11. Alternative drafts. The system will also have the ability to store alternative drafts of either straight text or text complexes.

12. Search. There will also be a word search facility, with which the user may look for the occurrences of specific words and phrases, tally them, index them, process them, etc.

13. Complex Connective Structures. Such connective facilities as super-footnoting, the ability to attach text chunks to an ordinary text and have them available under many different categories; parallel texts, where ordinary texts are linked side by side so that they may be compared or their relations noted; array formats, where other types of information may be stored in matrices and graph structures; and a cross-jumping facility, enabling the user to jump from a given part of the text to any linked part of the text.

14. Arrange and Printout. The system will be able to sort and arrange the materials it contains, and produce printed reports on demand, even interactively with a user. This will make possible catalogues concerdances, quick indexes and annotated manuscripts with some swiftness.

The Xanadu facility is meant to fill all these technical needs, and yet fill human needs as well. It is intended to be operated on an open shop basis for whatever purposes the company selects. "Open shop" means that the user will handle and direct the machine personally. This means that its elementary operation must be learnable in a very short time (a few minutes), and that it must be appealing to users. Indeed, the sytem must go out of its way to be appealing to users. It is safe to assume that persons of literary or editorial background will be among the most cantankerous customers ever to deal with a computer-based information system. In the confrontation of computer and litterateur it is the machine that must bow. All traditions of computer usage will be swept aside, in deference to the user's sensitvity about punctuation, capital letters, type fonts, and his presumable distaste for gratuitous numbers, and the arbitrary assignment of ordinary letters to strange new meanings. Our new conventions will be tasteful even to bookmen; but even these of communication between the user and the system will be subject to revision or redesign, in order to please the user or speed them up to his desires.

By the same token, the system's operation, physical plant, and surroundings must be designed for the user's comfort, and the facility should go out of its way to create communication facilities desired by particular users-- for instance, pushbuttons labelled, arranged and illuminated in a specific way because a user says he needs them, or a little "beep" responding to his actions.

Customized hardware may be fitted into a standardized console. That is, the overall response characteristics of the console are given, but users may request customized hardware such as special pushbuttons or pointing devices to readily communicate with a standardized interface.

While the system will be unlike any in existence today, it has many relatives. It is expected to have many of the good features of Engelbart's AHI system, SDC's SURF system, and Licklider's SYMBIONT. It subsumes INFOL, the information retrieval language developed by Olle at the Control Data Corporation, which is particularly concerned with the categorization and ordering of informational chunks, and their coding, as well as DOL, the Document Oriented Language, proposed by Mandalay Grems of Univac.

In addition, it subsumes in general function a large number of the kinds of work with texts now being conducted in scholarly settings. As a fair indication of this, we may consider the computer programs announced as available in the second issue of Computers and the Humanities. (Of the 51 programs listed, about 27 have their general functions included in this system.)

The system as presently designed will have one feature not currently available in any information system anywhere: the provision for the spin-off of alternative arrangements and drafts of anything; for their gradual shaping and the postponement of final deciding between them.

XANADU FOR BEGINNERS

At the beginning level, the Xanadu system is first encountered as a ~~looking into~~ screen upon which any part of the text may be shown, in both upper and lower case and with some quality of image, and where the user may perform editorial operations upon the text swiftly and with ease. These operations correspond to the conventional operations employed by a proofreader, such as the deletion of material, the insertion ot new material (which may be typed in on a keyboard), the moving of sections around the page (ordinarily signaled by a long horizontal curlicue), and the division of the material into lines and pages.

In other words, at the beginning level, the user will have hardly anything at all to learn. The standard editorial operation, or whatever operations exist within a given workspace, will be easy to learn. These functions will be initiated by manipulating pushbuttons or "virtual push buttons" (glowing spots on the screen with labels as to their meanings). The beginner on the Xanadu system will also have the opportunity of working through these particular work spaces and file disciplines. The basic editorial operations can be learned in minutes; the functioning of a given workspace in perhaps half an hour.

ADVANCED XANADU

While Xanadu at the beginner's level is rather like an extremely fancy piece of paper, advanced Xanadu permits the user to create, as it were, vast information structures hanging in space, connected any which way. The arrangements and facilities available at the advanced level (tentatively called "the Works"), will enable the user to determine the availability and manipulations upon information, including texts. The advanced user may define information types and create branching structures of response.

The advanced user deals with items, relation and clusters. He may create new entities composed of these objects, and he may direct the machine to create new functions, ways the machine will automatically operate upon data stored in the various formats described. He may define them generally, in terms of types of information and types of choices points, or specifically, creating a hypertext.

~~The advanced user deals with items, relations an~~

At this level the user will be able either to create complex or compound text structures (hypertexts), or himself to create ~~complex or compound text structures (hypertexts), o~~ workspaces for particular uses, arranging choices and carving screen formats arbitrarily according to his own designs.

By "workspace" is meant some arrangement or division of the screen, some particular way of having given informational materials stand by, some particular ways of arranging information of a given kind that will be taken as standard for a given purpose, and whose consistent use the system will enforce. He may create arrangements of particular information within these information types, choice points (and specifications of how the choice is to be made -- whether by push buttons, virtual push buttons, or other means), arbirary screen formats ("carving" the screen into separate sectors, not necessarily rectilinear), and functions -- collections of operations upon information and information types made available within the Works, which he wants to accrete into compound actions that may be selected or called arbitrarily.

This advanced language, too, will be made no harder to use than necessary. Although the machinery underneath which is to implement these functions may be technically called a computing "language," in the user's encounter with it there is little vocabulary or syntax to learn; as much as possible is relegated to screen pointing, virtual push buttons, and discursive explanations available through the screen.

XANADU FOR TINKERERS

Advanced Xanadu operates within the system program, the "Works." Below that level, it will be possible to add to the system in machine-language coding and programs compiled by Fortran or other compilers. While this addition will presumable be a continuing aspect of the system's administration, obviously few persons may be allowed to tinker in this manner. However, the design of the system should be such that additional functions for particular purposes (or to speed particular operations) should be easily addable within a user language. This is the Advanced level.

We may think of Xanadu as making possible "setups" for different purposes. Someone we may call a "designer" creates information types, functions and restrictions that are passed on to the user at a lower level. However, the user is permitted all the facilities that the previous designer has not forbidden to him. He too may now act as a designer, configuring or restricting the system that is vaiable to the next man below him. In this system the hierarchy of users is to be a matter of degree, and flexible.

SOME DETAILS OF THE PROGRAM FACILITIES AVAILABLE

These design functions are accomplished by the user in a user display language which permits ready control over the creation of files, interconnection of presentations, the making of screen formats, the creation of custom workspaces and work disciplines.

The user may declare his information types freely, creating as many different types of information as he wishes. The types of information need never be in coded or truncated form to the user at the screen, however, since any workspace can be directed to display (or have ready) full titles and explanations of anything.

The designer may declare his relations and connectors freely, again using as many connector types as he wishes. The meaning of these connectors may be assigned by the user, and it may be <u>changed</u>. Any possible arrangement of discrete information-- strings, vectors, arrays, trees, or graphs-- may be created, stored and displayed. (We may regard ordinary "straight" text-- ordinary paragraphs and chapters-- as being simply a special case of information which does not require this apparatus. But the system cannot be designed around this exception.)

On the most general level, we may regard this system as permitting the designing user-- the user with permission to create things in the "advanced level" user language-- to create a number of different things.

1. He may have <u>compound displays</u>, wherein several different parts of the text file may be brought to different segments of the screen at the same time.

2. He may create <u>workspaces</u>, that is, systems for the coming and going of different information types under specific circumstances, and responding in particular ways to what he does.

3. He may create work and file disciplines, that is, enforced procedures of working and filing.

4. He may create hypertexts, that is, structures of text passages with branches among them, that the user may traverse freely (or with restrictions).

5. He may even create hyper-workspaces. By "hyper-workspace" I mean multi-dimensional workspaces in which hypertexts and their variants and their indexes may be handled and compared.

The designer may leave options to the lower-level user; for instance, the ability to rearrange screen formats. The degree to which the early commands for a given setup restrict the later user are at the earlier designer's option. The "language" of the system at this "advanced" level by which these things are accomplished, will not use code-words or letter mnemonics, but rather-- as a matter of literacy and perspicuity-- whole phrases and arrows. The designer will be able to employ graph structure input, that is, the drawing of arrows on the screen, not merely for choice points among specific text material to be read in hypertext form, but also among choice points to be made available in his own facility or file discipline, and among all parts of the workspace which he is creating. Various other spatial mnemonics may be employed for swiftly arranging components and statements of the user language.

The designer may create screen formats, "carving" the screen into sections of any rectilinear shape. These sections may hold static information, moving scrolls, linked information that may come up, temporary memoranda, or any other data that may be defined within the setup. Such formats may be regarded as compound displays that draw materials from various parts of the contained file. (This may involve manipulating multiple pointers throughout the file, keeping track of the beginnings, ends and various other information about pieces of text garnered from all over the mass memory.) The designer may direct the system to jump from one display format to another on branching structures of his own design.

The choices and branches within a setup may be presented to the user as "virtual pushbuttons," which the user touches with a light pen or other pointing device. (However, when setups, workspaces, and operating conventions are sufficiently well-defined and accepted by users, these virtual pushbuttons may be commuted to a set of real pushbuttons in order to speed up interaction with the computer.)

The system will have features corresponding to the usual proofreader's marks for test editing-- such as insert, delete, move. In addition to these conventional operations, operations which have been found useful in other systems are also included. One of these is the "copy" operation, which allows a word or a phrase to be simply copied from one place on the screen to another place on the screen. Another of these is the "hop-to-link" operation, which permits the easy connection of materials in a simple text so that it is fairly easy to find all the parts which need to be related and understood together by the user. (All these functions are borrowed from Engelbart.)

A number of display modes will be available immediately to the designer. This includes blinking (any part of the text, the specific words, and so on may be made to blink), "scrolling," where an area of text may be moved up and down beneath a "window" on the screen. "Streamers" will also be possible: single lines of text emerging from some point in the screen and sliding leftward like the headlines going around the New York Times Building. Such moving displays may be put under a throttle control. ¶ Two or three different type faces will be available routinely. However, memory space restricts this somewhat. Presumably the usual faces available will be regular, italic and bold. However, foreign languages, (especially Russian), and mathematical notations will be available from the disk; albeit after a certain delay. New type faces or special characters may be entered into the system as dot matrices, and used in text display thereafter.

The user will be able to request justified columns or unjustified columns in displays or printouts; these would be calculated from input widths specified for each separate typeface. (The routine lookup facility should permit the input of numbers specifying the width of actual typefaces. This will permit the system to calculate justified lines, assuming that the width of its physical type slugs are known. (It is not necessary for this purpose actually to change the typeface on the screen from one Latin upper-lower alphabet to another.)

There will also be provision in the system for the use of "literal" material, i.e., pictorial matter stored as a system of instructions for the display. A reduction routine, with adjustable resolution, will translate light-pen or Grafacon input into straight-line segments. Larger two-dimensional pictures may be entered by light pen and modified, later to be retrieved through normal file branching. It will be possible to enlarge or shrink these graphic images, rotate them, and "carve" their sections or parts (to save, discard or combine) in much the same way that words and phrases may be excised and moved. This facility will be of use for the preparation of filmstrips and other graphics layouts, as well as experimentation with typography and direct-to-proof output.

Certain automatic counting, numbering and limiting facilities will also exist, permitting the system to count the number of words in a document, count specific words, show lines on the screen (or print lines on the printer) which are of certain length, number the pages of printout, count the appearances of specific words, and so on. The same word-limitation or size-limitation facility may also be used to restrict word counts or character counts under specific file disciplines (for example, in the imitation of commercially available systems for computer-assisted instruction).

The system will also be able to put items in numerical order and alphabetical order, and (when such items constitute one "type" of information linked to various others), permit the printout, in catalog form, of all the linked material. This will permit the creation of indices, catalogs, and so on. The system may perform word and phrase search within a file, presenting or printing out various kinds of information-- such as distances between occurrences of specific words-- as required. Indeed, this searching may take place independent of capitalization of words, italics, bold face, etc. (Indeed, it is intended that the system shall even be able to find words despite what we may call Salinger underlining or italics, in which a part of the word is italicized while the rest of the word is not.)

There will be a facility for "don't care" searches: searches for words and phrases some of whose components are not considered. Whether these "don't care" fields may be of variable length is not yet clear.

The facility will exist for a breakthrough to Fortran, permitting the writing of numerical programs through the facilities supplied ordinarily with the computer. This will be of use for statistical analyses, mathematical evaluations, and so on, as required. However, this particular facility will not be adapted to ~~the~~ swift change in numerical programs. If that is desired, it is possible to lease for the PDP-8 ADR's ESI language, a variant of the Rand Corporation's JOSS System, which permits immediate response by the computer to all sorts of numerical queries of rather high complexity. This is an interpretive system which would reside on the disk.

The system will also be able to turn things on and off-- a "ding-dong" facility-- and this control will be accessible to the user at the setup design level. (This ding-dong facility may be combined with the simple graphic routines, for instance, to create rudimentary animated films without additional programming. An animation artist could create simple shapes, move them between frames and change their sizes; and the machine would take all the pictures when so directed.

There are a few more odds and ends of peculiar features. Within a sufficiently dense connector structure, it may well become necessary for priorities to be established among the various response times of the things the user wants. (It may be possible to specify priorities of connectivity which are to have relative priority, by drawing "iso-connective" gradients upon a rough map (planar graph) of his file in much the way that areas of equal temperature and pressure may be drawn on a weather map.)

It will be possible to use what may be called "detachable relations," or, more precisely, the detachment of the relations among items or chunks of texts and the creation of new relations among all those same items. (An example might be the following: suppose a novelist, trying to establish a chronological sequence among events in a novel, takes all the single events in pairs, and says, "This comes before that" for a large number of such pairs. He may then ask the system to show him the sequence he has created. However, the system may then inform him that he has not created a consistent sequence. Rather than be constrained to remove some of the paired relations he has established, he may redefine the relation, perhaps as "lattice of flashbacks".)

THE ANATOMY AND MECHANISMS OF XANADU

The system program, or augmented operating system, is intended to take care of all the functions and activities that have been described above.

This system program is to handle many functions concurrently. These include 1) arranging and maintaining the file of documents in the system in as elaborate a connective structure as desired, and modifying the file as required; 2) maintaining and modifying the display; 3) ~~the~~ handling ~~of~~ other input and output. (Whether the latter two functions should be handled by a common report generator is an interesting question.)

This system has essentially two kinds of output. These we may call fixed sequential output, the continual output of changing materials which have been prepared in advance; and responsive branching output, the output of materials and displays which are subject to continual change according to the decisions of the user. The former is for the line printer and such activities as video typesetting; the latter is the continuing function of the display/file subsystem.

It is intended that these two forms of activity, with their different output rates, may be interleaved by the system for efficient utilization of the whole.

We may perhaps think of the Xanadu system program as having three parts: the part that communicates with the user through the creation of displays on the screen and the decoding of messages from him; the part that makes note of all the connections within the file; and the part that operates on the file and rearranges it. These program parts may with justice be called Sexus, Nexus and Plexus.

There are many different possible designs for the inner structure of such a system program. In principle the system program should have an inner core which is as simple as possible, for the sake of cost and reliability; but this is a matter of programming elegance rather than apparent complexity of what the machine does. We may think of this inner core as a sort of pepper grinder: the simpler the motions involved, the less there is that can go wrong.

The conventional way to design such a program is to figure out all the different sections that are needed, then map out all the transitions that must take place among them. The more elegant approach seeks to create a compact central mechanism to handle all these arrangements and transitions. We might draw a comparison between a banquet table at which the participants specifically request all the different dishes from each other as individuals, and one where all helpings are taken from a lazy susan.

There must be underlying, primitive functions or operations which the user may assemble and configure to do the things that have been described. The selection of such primitives is a matter of judgment, with some arbitrary aspects. The intent is to choose operations which will allow all the intended purposes and then some. However, they should not require that the user concern himself with features that can be handled automatically (such as arranging text on the disk) or may contain hidden dangers (such as allowing the ordinary user to write machine code).

It is anticipated that the primitive functions will be concerned with the establishment of relations and file connections, the creating of output lists, the manipulation of pointers to and from disk, core and screen (actually screen buffers in core), the traversing of graph structures and the comparison of items, and the analyzing of relational clusters that may occur.

A (next page)

So will user manipulations and modifications. Under ordinary circumstances a text or file may be stored as original input plus (and minus) charges, available in sequence (and even with time of day, if needed). While the updated working file will be organized more efficiently, it will be possible to look at earlier materials by peeling off charge operations (or perhaps re-assembling up to a certain charge level). Such reconstituted prior versions may serve as raw material for spin-offs.

This system will have certain peculiar operations and functions which deserve special mention.

First of these is what we may call the "representation function." This function declares that the separate members of one cluster of units -- such as a list, collection or network structure -- is connected, unit-by-unit, to some or all the members of another cluster. This second cluster may or may not have the same number of elements as the first. Its paired elements may be in different order; and there may not as yet be any basis for pairing the members of the respective clusters which are to be associated. The second cluster may not even exist yet.

This "representation" function is intended to be a basic method for all indexing, for adding new information types to the file, for creating footnotes, and for keeping track of relations among separate versions.

A second specialized and important function we may call the "spin-off" function. This permits the creation of "separate" versions of a document, hypertext or workspace. These may be independently modified and rearranged, but will remain connected part-for-part to the original for comparison.

Whether the representation and spin-off functions should be coded at the machine level, or as compounds in the user language, remains to be discussed.

Most operating systems have only two levels of program operation, "environment" and "program execution" (under the control of the environment). This system should have a gradation of program levels. These will range from what we may call "wide open" (the bare system, treating any user as a designer) to "completely restricted" (where a user has been allowed by a designer only a narrow variety of options, and has no access to the system language). This is the "accretive" approach, mentioned earlier. Numerous levels of designers may contribute to a setup. The designer, and (if permitted) the ordinary user, may use this system language to create file and connector types, hypertexts, workspaces and so on. He may also create compound operations of a wide variety. These may take the form either of constraints ~~xxxxxxxxxxxx~~(to operate within his own setup), or of utilities and augmentations -- compounded instructions -- which may then become available to other users of the system.

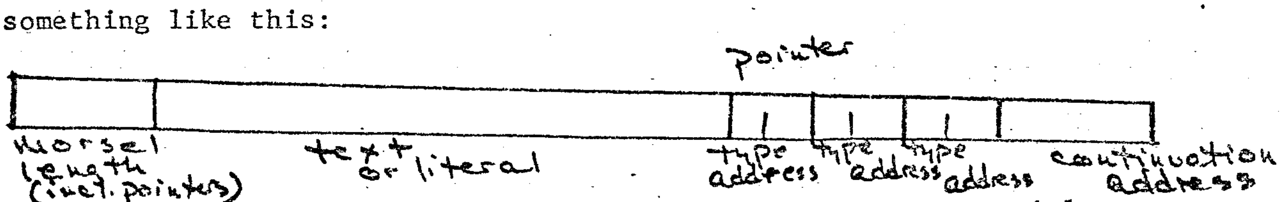
Presumably programs in the user-language will be stored as character-strings in the same kibbled format as text and other material.

It is anticipated that the general form of storage adopted for this system will be unusual. It will be in the form of what may be called "kibbled text" or "kibbled data." The information will be broken into short lengths which are distributed or scattered irregularly around the storage media and connected by pointers.

It is expected that all the pointers, compound entities, categories and clusters invented and used by the on-line user will be stored in the machine as some form of <u>relation</u>. This will lead to a smooth and simple algorithmic treatment, deriving from the relational calculus, group theory and graph theory. It should also tie in well with the use of visual inputs rather than more conventional alphameric languages. This is to make possible the containment of anything at all, records and items and titles of any length, and pointer structures to any degree of complexity. Length of individual items is unlimited; while their actual machine storage is in blocks of fixed length, these are linked together within the system so that the user need not concern himself with any limitations of space. (The unrestricted connections and item lengths are possible only, of course, if the user has not been restricted through the creation of some restricted workspace.)

The user is unconcerned with the physical location of information, although he may be able to specify response priorities, designating what things he wants quickly, which will affect the use of space.

We may visualize a single "morsel" of kibbled information as looking something like this:



For input, in screen buffers, and sometimes on tape, materials may be stored in "printout" form, that is, in some way that matches quite neatly the way in which they would be printed on a page. Note that this is not the same as the distinction between ordinary materials and materials in <u>canonical</u> form; canonical form means that a thing is finished and done, and may not be changed. This form may be of interest when dealing with already-published materials which are being held for reference.

The system's real-time clock will permit some form of time-slicing. This will facilitate the machine's transfer of attention among separate users and input-output activity, as well as providing a hedge against endless looping.

The user will also be able to search for specific sections of text, as described earlier and request combinations of things he has sought according to any Boolean function desired.

There will also be a text compression mode available for large, redundant files of text. This will permit storing words, not simply as sequences of letters of the alphabet, but as numbers which refer to large tables of these words. For corpuses of a certain size and redundancy, this can be a useful technique.

It should be noted that an interesting feature of this system, which distinguishes it from other "real-time" systems, is that delays are tolerable. While they may be annoying, delays are not likely to result in great queues of undone work piling up on all sides. In this important respect it is not like an airline reservation system, and may be much easier to program.

THE HARDWARE OF XANADU

The proposed implementation of this system is to be on a small, general-purpose, high-speed computer. The best machine for this system is the Digital Equipment Corporation Display-8. (This is now the advertised name for the Type 338 Subroutining Display, which incorporates a PDP-8 computer.) The Display-8 is the most powerful display console commercially available; it was designated one of the one hundred most important industrial products of 1966 by Industrial Research Magazine. The first of this line was recently delivered to MIT, and others have been ordered by important research centers, including Bell Labs. This is to govern two consoles, each with keyboard and light pen. The 338 is a peculiar machine, with a powerful display controller that follows a program of its own. Its program is stored in the memory of the PDP-8, whose pocket it picks (by cycle stealing through direct access to memory) at intervals determined by its particular task. Otherwise the PDP-8 is free about two-thirds of the time.

This machine is equipped with a unique character generator, the VC38, which permits upper and lower case and a variety of fonts, including italics, the Cyrillic alphabet and mathematical notation. In practice the system would be limited to perhaps five or six unless a delay is tolerable. This feature is available nowhere else. A restriction of this console is that it is able to present without flicker only about 800 characters of text, all told, on all scopes. This is the reason for the "primary/secondary" division of work between the two consoles.

The principal form of storage is to be on a large disk file, one holding several million characters. The Burroughs disk file is preferred because of its high reliability and fixed head-per-track system, which makes it unnecessary to position an arm before reading data. The model that appears best for this purpose is the B475, whose size begins at about ten million characters, and which can be expanded with up to four more units of this size.

In explaining or demonstrating such systems, it is a strong temptation to keep saying "Bingo!" This stresses the immediacy of the response, the next presentation upon the screen. It is anticipated that Bingo in most cases for this system will be about 1/10 of a second, for a single terminal, This is because the disk is fast. However, for complex inter-connected materials which require following, say, half a dozen pointers on the disk, Bingo may take as long as 1/2 of a second. Of course, searches and computations will take considerably longer. It is anticipated that we may be able to treat magnetic tapes and disk files as one large overall form of storage. That is, materials which are not on the disk may be found on the tape, regardless of the complexity of the interconnections within which they are stored. This permits the effective extension of the disk by millions of additional characters or cells. Of course, such a "swirl" through tape will considerably increase the Bingo coefficient. (However, tape swirl may take place concurrently with other activities.)

Because it is expected that new ways of working will be created in unpredictable ways by the users, it must be possible to modify the available switches and pushbuttons as needed.

This means modifying somewhat DEC's original pushbutton idea, so that new sets of pushbuttons, to a large number, may be established for fixed-function and established-workspace operations. For this reason the current plan includes add-on six-bit input registers (custom-designed by Ann Arbor Computer Corp.) for typewriter keyboards and miscellaneous, unpredictable accessories. This input is intended to aid in the standardization of the Xanadu terminals. The 338's built-in pushbutton bank does not accord well with the philosophy of making user accessories correspond in appearance to what the user is thinking about. It is adapted to programmers, not laymen. Hence, this pushbutton box will be de-emphasized. Similarly, because the machines has extra address bits for the display, it is often spoken of as having a display "lake" of which only a part may be viewed. This approach will probably not be useful to us, except perhaps for specialized graphics production. However, certain other machine-dependent features ~~will also be incorporated into the display;~~ should be exploited, particular the hardware size control (three sizes) and brightness control (seven levels).

PROGRAMMING PLANS

In the programming of this system, it is planned to use a form of "bootstrap" construction. An intial program, either created for this purpose or obtained elsewhere, will permit the writing of programs, flowcharts and other documentation on the screen itself, with text editing and linked documentation. It will also make possible linkage among program segments and among document segments. Such an approach should substantially expedite the work. (We might use Magnuski's DOC program, ADR's editing program, or a number of other things.)

EXPANSION

The system as currently designed has a large number of options for later possible expansion.

The use of a turnaround console will permit users to come and go without interfering with one another, a very important feature. It will also permit various minor activities to be handled by secretaries in "secondary" mode, on a console with reduced capacity, concurrent with full operation of the other console.

More core memory may be added, permitting faster responses and facilitating storage-tube consoles (see below).

More disk modules may be added; each of these contains approximately ten one million characters. The main disk controller will handle up to five such modules, a total of 50 million characters. (Or more, if condensation is employed.)

Additional tape units may be added. Two tape units will be better than one, so they can be used with the two consoles and delay can be lessened in loading and unloading disk.

The use of storage tubes, in most respects of their behavior like the ordinary Xanadu console, should be experimented with. It is a peculiar feature of the Xanadu configuration described so far that it should be possible to add additional storage tubes, operating independently of the main console. The problem is one of memory (both disc and core), rather than of display capacity, since to lay a display out on a storage tube takes only a fraction of a second, while to maintain that display on a volatile display tube the picture must be sent out some thirty or more times a second to avoid flicker. One of those thirty frames, diverted to the storage tube, is adequate. How many users could employ the system gainfully through storage tubes is not yet ascertainable; for some applications five or six might share it.

Finally, the addition of more 338s to the disk configuration should have powerful advantages in expansion, and for four or five volatile displays would be the easiest way to expand. One is the obvious advantage of requiring no changes in any of the programming. The importance of this capacity should be obvious. It should be feasible to put some three or four 338's around a full disc compliment of 50 million characters.

(There should be a word on the nature of the hook-up between 338's and disc. 338's can communicate with each other via the disc, certainly at least, in the swapping of data; however, it may well be that fast core-to-core data transfers are also desirable. In this case, particularly for symmetry of the configuration it might well be best to establish core-to-core communication "around the circle" of 338's.)

Additional 338's, however, would not need many of the facilities desirable on a flagship machine we have been discussing. In particular, parity, the extended arithmetic element, the loudspeaker, and possible even the ADC and DAC, may not be required. It is unfortunate when discussing lookalike units to distinguish certain units as in some way special, but in this case it would seem appropriate.

It will be noted that beyond the basic system described here, further additions require no modifications or special adjustments. This applies to core, tapes, disc, storage tubes and additional 338's. In this sense, the configuration described here is <u>complete</u> in a way that any smaller facility could not be. The hardware above this level is modularly expandable.

It should be pointed out that the wiring of the system should prevent the establishment of consoles at least one floor above and one floor below the basic configuration. Wire length is the crucial limitation. The distance between scope and 338 may not be greater than 20 feet, but that can be 20 feet in any direction. Thus it might even be possible to have consoles two floors away.

SALE: XANADU AS PRODUCT

While it is anticipated that initial uses of the system will be proprietary and confidential, it is well to consider the marketing possibilities for such a system. While our firm would of course not be involved in the sale of hardware, (the Ann Arbor Computer Corporation would probably be quite willing to take care of all this), the deal would be as is usual in such cases, the leasing of programs. Our copyrighted system ~~might be~~ program and documentation might be leased (at, say, $50,000 to $100,000 a year) to various places at which documents are extensively reworked and manipulated. ~~But~~ Various document mills, such as magazines, foundations, movie studios, the White House, perhaps newspapers, and universities wishing to keep their most productive faculty, would be prime markets. (Whether this could be marketable to others in our field is another question.)

This system might also be used as the front end of Management Information Systems, as its capabilities are extensive and unusual. Such systems have previously concentrated on computational response, and their designers have tried to anticipate the data structures and output forms that ~~will~~ will be needed. This is the wrong approach. XANADU, with its relentlessly versatile reconfigurability, will be continuously adaptable to any new uses and requirements. (And it is quite likely that competitive systems that would appear-- say, Honeywell's HONEYDEW and IBM's ZAN-TRAN-- would fail to grasp the necessary depth of this ~~reconfiguration~~ reconfigurability, and fall short.)