Interactive Systems and the Sign of Virtuality

In Part I, we considered some nice examples of highly responsive systems. The reality of their implementation details is comparatively unimportant. What is important is the design of the conceptual structure and feel of a system; we call this its “virtuality” as distinct from the (unimportant) reality.

In this concluding section, we consider some more design examples, and endeavor to find the right principles on which to base the design of interactive systems in general.

A Complete System

In one design, the Funny-Face System, I have endeavored to show that one simple, overarching control structure can be used for a complete personal computer system—including word processor, scheduling system, graphics package, bookkeeping package, typesetting and layout programs, etc. I do not wish to imply, of course, that this is the only way to organize such an integrated system; merely that this one interests me.

There are four basic controls. These are the only controls. They may be understood quickly in a brief demonstration, but in fact the further ramifications of their interaction may become clear gradually.

The controls we call up, down, around, and flip.

I would marry these to the Radio Shack keyboard as follows:

Up and down are the easiest. The user...
We may call this a latching tree. From your viewpoint, you can see the trunk, up. If you go up, you get to the top; if you go down, you get to a path of already latched, or chosen, selections.

How do you change the selection of the node which is found? Do you do this by panning around, which selected in turn each of the different alternatives below. (I call such a circular succession of choices a "node." This term is used to describe between any two places on the tree only a few particular selections and not something like up, up, around, down, down.

That's essentially all there is to it. What you have seen is what the basic menu does. I have left out showing how the different parts combine, so that, for example, the graphics tables used with the scheduler produces animation, or the scheduler used with the word processor programs a magazine layout.

I would point out certain other features, however. One is that there are few very steps between selecting an item and the user going back and forth between them gets into a rhythm. Faster methods would be in reality not simpler. Another aspect is the system's uniformity of replicative structure. You go anywhere with confidence, for the structure will hold. (It does become quite in memory, however, at the bottom or execution level.)

Some people tell me they'll rather have an interpretive command. This is all right. That's a matter of taste. Other critics say this system lacks the generality, which misses the point. It is simple, easy to learn, and integrated. You can't beat it. And the funny facts are good for a laugh.

THE XANADU HYPERTEXT ENVIRONMENT

The Xanadu Hypertext system, toward which I and can work for some twenty years now, is intended as a super document library and annotation system, among other things. We may also think of it as a new form of writing and publication in itself. The Xanadu system is planned as a network of storage computers. McDonald's file contains thousands of addresses around the country. By dialing into your local Xanadu station, you may get any part of the whole network, and you can browse, and view, and edit this information, and print, and otherwise interact with it. You can access the system from the local computer, or from anywhere in the world, if you have access to the Internet. This is a powerful, new computer, for reasons which will become obvious later.

While most of the Xanadu work has gone into problems of its implementation -- particularly, algorithmic analysis -- the system's emerging value is the system's emerging value in the emerging value of the system's emerging value, which I will now describe.

Everything stored in the Xanadu system can be a document. A piece of text, a picture, a movie, a sound, a font, a range of notes, or -- almost anything is a document.

Any document you want comes when you ask for it, if you are entitled to it. A document is private or public -- that is, published. Any user may call up a public document instantly, as well as his own private documents or any other private documents he has permission to see.

LINKS AND WINDOWS

Links may be put anywhere in an document. Links, like footnotes or marginal comments, permit a user to jump up related material at any time -- and come back from that other material when he likes.

Free-form, non-sequential writing of any kind is also called "hypertext." It is made possible by the links. But the versatility of general hypertext would take a book in itself.

An important type of link is the window. A window is any place in a document which is referred to as a "hole" in one document through which shows a part of another document.

CHANGES AND VERSIONS

Not only may an author store a document, but an author may change it. If the user chooses, write or rewrite the document on a local computer, and show it to others who wish to see a version of it stored. The Xanadu system does this essentially, even using a incremental cost, since our data structure and algorithm easily assemble parts of a given version, and a version of an edition without ever bothering to assemble the full consecutively stored version, which would have been slow.

Thus the user has access, if the materials are available, to any version he wants to use. If there is no version of the document in the current version of the document, the program will prompt the user to specify.

Not merely concurrent historical changes are possible, but alternate versions of the same thing, generated at any time. Thus a document may be "guaranteed" for different types of readers, and these different versions stored at a low common denominator.

The user may ask to see any given piece of text (or other information in any version of an earlier version of any text or at any previous version of any text). The VersaWriter graphics table makes it possible to view the document in its entirety, and draw drawings with your Apple computer. It can be read in quality to graphic bit maps and digitizers costing three times more. VersaWriter is a digitizer and software package which presents a new approach to further graphics. It com- pensates of a master plotting board with a clear plastic overlay. Attached to this board is the drawing arm, which has a magnifying lens with a cross- hairs at its end. You simply place any graph, picture or drawing up to 8½" x 11" under the plastic overlay and "trace" it with the drawing arm. As you trace the drawing appears on the video screen.

The superior software of the VersaWriter enables you to do much more than trace. Immediate commands include: color choice, color, color size, color, color type. Axis/Distances this program allows you to calculate distances (or perimeters) by establishing a measurement unit (of your choice) and tracing the shape or map route with the drawing arm. Areas of figures are calculated in the same way this includes irregular and open figures. A very special calculation program is also on this software disk.

A secondary software disk contains the VersaWriter demonstration programs. For more advanced use of high-res graphics, there is a skeleton program which contains the guts of the VersaWriter. The VersaWriter is a sturdy peripheral device which plugs into the game paders I/O port- the VersaWriter does not use a card slot in the Apple computer. Also, the VersaWriter is not subject to the grounding problems and strong magnetic field problems of other, more expensive, high-res graphic devices.

VersaWriter requires an Apple II Plus or Apple II Plus (or an Apple II Plus), and a least 12K of memory, and a disk.

VersaWriter comes complete with software, plastic overlay and two soft disks of software. Price $25.00. postpaid in continental USA. VersaWriter has a 90-day warranty on parts and labor.

Credit card customers include card number and expiration date of your VISA, Mastercard, or American Express card. No C.O.D. Bankcard customers may order toll-free to: 800-631-6112. (In New Jersey 940-0455), Dealer inquiries invited.
A link must to a certain part of one version of a document may be automatically followed through to the same matter in any other version of that document or in its previous incarnations or in other publications that document it.

We believe that this "merging" facility, of linkage across backdraft and alternative versions, when a central program of text systems -- that of transreferring any parts still being worked on -- a problem which is chipped at and nibbled at everywhere but in fact dealt with in ineffectual ways.

**FREE LINKING BY ANYBODY**

You may create a document that links to any other documents, if they are public (the who-what versions must agree to this in advance).

You may, in your document, windows to anybody else's public documents. (Once they get the privilege when their part should, they should be pleased.)

This is how we handle marginal notes tied to its operating note, it is automatically put in a new companion document, your document, which is permanently linked to the document you have annotated.

(Here is a combination "document" that also figures to allow and secure any public document in any way you like -- since the annotation is in a private file of your own that points to the intact original.)

**COPYRIGHT**

What of the copyright problems? Ask our solution is simple: you use the system, you are continuously paying small increments of royalties to copyright owners. There are modest amounts, the same for all users: for instance, if we can supply the service for two dollars an hour at 50 characters per second, the total royalty runoff will probably be about five cents an hour. This is divided among the copyright holders in proportion to how much you used from each -- slipped very finely.

What keeps people from making copies? Nothing, since terminals and under the control of individual users; but since everything is still stored on the system and available instantly, the cost and inconvenience of making and filling private copies will be often seen as superfluous.

**OVERVIEW: THE XANADU SYSTEM AS A VIRTUALITY**

The above description specifies a general and powerful facility for business, literature, correspondence and digital storage of all kinds. As such it represents a cohesive and unified virtuality which has been thought about and reworked for years. Its appearance of simplicity and obscurity is due to the distinctive quality of a carefully wrought design: There are hundreds of ways to do these things, as experienced computer people well know; yet making the parts hold together cleanly, complement each other, and make sense, takes a very great deal of work.

**XANADU FRONT ENDS**

Of the functions described above, only a few are actually handled by the Xanadu service network; put this away and give that "hand of incandescent a version are really all that the Xanadu back-end machines do.

The rest has to be done, actually, in your personal computer. Marginal notes, for instance, require making a companion document out of your margins, for instance, and declaring it and putting it away in the network. Most users will also want to keep track of how they have been jumping among various documents and activities. These necessary functions belong in your own computer.

Thus the "full" Xanadu system as we recommend it be used, entails a cooperating program in your personal machine that acts in these ways.

Thus full Xanadu service has two parts. The "back end" is the program Xanadu network, essentially all it don't store and fetch by versions and last.

But a high-powered terrestrial is needed by the user, to show the documents sent by the back end, to present the possible actions the user may take, and to translate those choices into the proper feed-and-send instructions for the back end.

This is of course the "front end." There are many possible ways to visualize and control the Xanadu functions: even before graphics or music are stored on the system -- and we welcome imaginative front-end programs of any design, even if created independently. The Xanadu project will, however, offer some guidelines for front-end design.

If you choose to use the back end at all, it works in some other way, than is your creation. As a result, we demands that you encourage the way we see as desirable mode of operating, in which you may be using various trademarks to software vendors who wish to create cooperating front-end programs.

Given the overall virtuality of the Xanadu system, there are countless possible ways to summon, visualize and control programs on screen. All of these are valid and welcome. To give some ideas of the possible varieties, I will discuss two very different Xanadu front ends.

Since these are highlights of the two front ends, no attempt will be made to show all the functions, reconcile their different emphases, or intercompare them.

**THE XANADUX™ FRONT END**

The standard Apple computer, usable by anyone in principle, is not so whole, and its capabilities, as a has a few conspicuous limitations. One in its text screen, only forty characters wide.

However, an Apple strength is an instruction low-res graphics. Two pages of hardware memory are dedicated to either text or low-res graphics. We wish to make use of this fact.

The Xanadu front end has been designed for fast and exacting use of the Xanadu facilities, as well as for ingenuity and use of its low-res graphics.

The system was, quite frankly, inspired by Star Wars, and shows how far you can go in playful and analogous use of graphic symbols.

One of the things a Xanadu user must be able to do instantly is ask exactly what he is looking at; that is, having jumped to something or wandered by degrees from the original activity, he ends...

**CREATIVE COMPUTING**

For all the CP/M® and OASIS operating systems

small business applications, inc.
3220 Louisiana • Suite 205 • Houston, Texas 77006 • 713-528-5158

Available for both the CP/M® and OASIS operating systems

CIRCLE 144 ON READER SERVICE CARD

**The MAGIC WAND is TM**

almost perfect.

We've been saying it for a few months now, and the reviewers seem to agree.

Until I saw the Magic Wand, if I were allowed to own one and only one editor, Word Star would have been it. ... My personal preference is for Pencil or Magic Wand for text creation.\(^{(3)}\)

Jerry Pournelle
On Computing, Summer 1980

The basic functions of the magic Wand editor are as easy to learn as those of Electric Pencil\(^{(4)}\). ... Magic Wand dominates in the area of print formatting.\(^{(5)}\)

Larry Press
On Computing, Summer 1980

Of all the word processors I have used (and that includes a dozen or more), the Magic Wand is the most versatile. The Wand has almost all of the features of other processors, plus many new ones of its own. It measures up to even the word-processing software running on the largest mainframe computers.\(^{(6)}\)

Rod Halle
Microcomputing, June 1980

The Magic Wand is one of the most flexible word processing packages available, and should be considered by any potential word processing purchaser.\(^{(7)}\)

Glenn A. Hart
Creative Computing, August 1980
The world created by a programmer or designer. This world has a certain structure which may be easy to understand or hard. This world is visible through different views allowed by the designer.

The World is what you’re really thinking about; the view is the temporary way you’re looking at it.

The distinction between World and View is crucial. The World is what the user is supposed to be acting on and thinking about; the View is all he really gets. (Controls are in a way a part of the View.) If the World is good, the View comes to seem real, natural, at hand under control. You perceive it, not the world, a hard-to-envision world that creates confusion and poor usability.

The system should have easy-to-use visual states and conditions, and preferably some kind of spatial orientation so that recently becomeremembered thing seems more like your mind.

The designer should begin by thinking about visualizing the ‘World’, not the View — the things the Views come later.

Principles of the World are the central ideas, the steps, the steps how you see it is secondary. It is important to acknowledge the centrality of World design, and consequently the importance of the principles you develop for it.

The designer creates a simplification or stylization of the original world. There must always be some reduction or style in the important thing is that these reductions or stylizations do not detract from the principal things you need to understand and control.

In transposing an old activity, the question is what to retain in the world and what to diminish as part of the View. It’s important, when you’ve started a program, to make sure all of the things you think you’ve got are part of the View also.

Any world can, be anything, any world can be on any world, whatever on whatever your mind was asking for bigger screens — but actually to ask for bigger screens is usually a copout. Ask for higher performace for higher performance or compatibility.

Patterncs and fades and scrolls and panel groups are there for and serving cues. Leave several things on the screen at once to remind you of what you’ve been doing, what you might be doing, what else is going on, and do any other current options.

The attributes in the user environment should feel much like the rules on a work board. Anything can be controlled by almost anything, talking or listening or typing or a keyboard. An interactive system should have very few things these few should have far-reaching and powerful use.

Mane the controls available and the

CIRCULAR CONTENT

WWW

THE WORLD AND VIEWS

An interactive virtuality is essentially a

THE BARTENDER

AModified Prayer: Suicide, Vodka Martinis, and much more by simply asking your computer how to do it.

STREET DRUG INDEX

A MUST FOR DRUG USERS AND STUDENTS

IN PATHWAY

12 oz of 50% vodka

LSD

1000 micrograms

Bartender: ((KID) 646.1919)

Drug Index: ((KID) 646.1900)

Be careful

SHIPPING AVAILABLE

FOR THE 90 ONLY

CIRCULAR 145 ON READER SERVICE CARD

EDU-WARE

WHO IS EDU-WARE?

We are CAL PROFESSIONALS FOR THE APPLE II.

At least an EDUCATIONAL INSTITUTE?

EDU-Ware is a small educational business in the development of pre-packaged instructional software systems to be employed with the microcomputer. We challenge those who may make similar claims to match these specific facts:

A management team trained in and focused on Instructional Development.

Programs that pre-test, pre-teach, and re-teach, provide practice, reinforcement, and test for mastery learning.

Instructional algorithms which inject independent inquiry while reinforcing those which are enough to make those screens come alive and make working at them a joy. Which is the real problem.

EDU-WARE

IN THE BAR

EDU-Ware recognizes problems for instructional materials that can make a student demanding too much— easier and more effective. Our software supports you in this effort because:

The employment of hypertext, double-target, instructional methodologies maximizes affectation of teaching goals.

Edo-Ware trains your computer into a powerful teaching tool, not a manual student

Interaction with Edu-Ware systems develops students’ self-management of learning.

When you or your school employs an Apple Computer and then acquire it in quality C.A.L. Ware, please write for product information...

Edu-Ware Services, Inc.

22583 Scenic View Rd.

SUITE 211

Redmond, WA 98052

212-340-0712

Edu-Ware Services, Inc.

22583 Scenic View Rd.

Suite 211

Redmond, WA 98052

212-340-0712

CIRCULAR 145 ON READER SERVICE CARD

CIRCULAR 144 ON READER SERVICE CARD
Virtually, cont'd...

desired functions. Menus should be used, rather than input languages or the fictitious "natural language dialogue," or better, yet, context diagrams.

Actions should be easily reversible and their consequences immediately recognizable so the user can back out of a mistake without being punished. (Compare this with the word processor/horror stories you hear all the time.)

Most important, the overall principles you choose for a system should be sweeping and have few or no exceptions. In order to clarify these issues I must consider the issues of both soft principle and soft clarity.

THE PHILOSOPHY OF SOFT PRINCIPLE

The following discussion has to do with the design of principles, which is in fact the essential issue.

SOME FAMILIAR IDEAS SOFTENED AND RECONSIDERED

We frequently consider something and ask ourselves: What are the implications of this? And one of the nice things about science and technology is that the implications tend to be clear and exact.

In many cases, though, implications don't follow clearly from premises. Those who want clear-cut answers become edgy or anaesthetized. The writer has been trying to find the exact implications of science. He has found many ways (ideas which don't seem to have exact implications) as well as people who prefer unclear situations, clusters in the humanities or "fuzzy studies.)

But some things are by nature unclear in implication. These include both cluster-concepts ("Democracy," "Womanhood") and design ideas ("Let's see, maybe it could fold back onto itself somehow..."

By tradition we often tend to talk of such ideas as improperly formulated or like better than others, and a variety may be valid.

Now there are a few soft design ideas, all at once. How do their implications fit together? The answer is indeterminate, since the ramifications of each could take many forms. How is it clear that the design search could be done carefully for the combinations of possible workable/whose variety and their consequence.

The "inspired" design of something familiar and precise comes, I believe, from realizing that many such co-implications of possible hardenings of the ideas. And the important guideline is: don't rush it. Don't take shortcuts. Don't assume that decisive planning down one aspect of a design will speed things up. It's like nailing your left shoe to the floor. If we think of design as the search for many possibilities, "soft design" is that which is sensitive to unexpected situations, conversons and harmonies. In short, don't be too sure of what you're looking for, and be ready to accept the ramifications of surprises.

Principles in Practice

Eventually, the soft design principles we have tried out most long have hardened into specific hard forms of computer operation. What should we do? Get them to like? Again tradition attempts to adapt against recognizing the test design signs.

The general principles of a system, one by one, should be habituated, be "consistent" according to lesser criteria than the designer may be used to. In particular, a design principle may be psychologically clear for people to work with, easily visualized or imagined, yet not reducible to any customary formal usage.

Indeed, "consistency" here takes on a strange psychological flavor: a thing is consistent if users think it is consistent and use it consistently—just as if we don't like it, like the double negative in Spanish. (We may call this naive consistency or soft clarity.)

Thus the final chosen principles need not be "logical" in the rigid sense of conforming to somebody's pre-defined notion of how things should behave. But working out in soft form, we study their fittings-together in great detail.

The designer should eliminate any background notion that the user must be like him. All too many designers reward the user for being like himself, the designer, or punish the user for being different in thinking differently. The objective is to be of service, not to clone yourself.

TECHNICAL TRADITIONS VS. SOFT DESIGNS

The design of virtuosity is essentially the design of operating principle. The design of principle, in turn, has to do with the generation and modification and inter-sculpting of soft principles.

The biggest design problem, though, is that the designer is quickly on a particular set of rules and arrangements. Technically-oriented people tend to see one or two principles and hang onto them through thick and thin, not percepting when it is time to rework the whole.

I have had the bitter experience, indeed, that only a small proportion of technical people are even capable of listening to this viewpoint. The soft design of virtuosity seems to be alien to technical training.

* * * * * * * * * * * *

** Note: Virtually, the word is used on the original JOT system, sometimes it is a consistent virtuosity, even though it is "untranslatable" to the modern paradigm of computer structure.**

104 CREATIVE COMPUTING
Virtually, cont'd...

Those who design interactive systems tend to be technically trained, and technical training generally promotes the background assumption that what you are working on is given and well-defined. Training in the arts and creative fields, on the other hand, promotes the idea that a work of art—be it a painting or a movie—is fluid, may take many forms, and will be revised or reworked over and over until it reaches a final state that may be wholly unlike its earlier stages. I believe this latter outlook is far more appropriate for the design of interactive systems.

CONCLUSION

Interactive system design is a field in itself, strictly unlike what is taught in any computer science major I know of. If I have not proved this point, I hope the demonstration examples cited here will at least provide some source.

(This is no claim that these designs are richer than any other; but rather that these are unified packages that feel right and is therefore interest. They represent local peaks in design space, in the sense that small changes would, I think, detract from their unity and clarity.)

These designs represent hundreds of hours of work, but the difficulties of the designers and the rough draft don't show. (That's part of good design and art.)

The art of designing things in general is very little understood. People think that something is well-designed if it is slick, stylistically unified, and if its controls look as much alike as possible. (An example is the "designers" audio equipment from Bang and Olufsen, shown at the Museum of Modern Art and copied everywhere, which everyone unanimously reveres every other control.)

This approach is wrongheaded beyond belief. (I think stereo equipment is poorly designed, and B&O the worst of them.) You do not build a machine that looks alike. You want controls that look and feel different. If you have a big round knob for the volume control, you should have a square one for the tone, a flat one for the pitch. There should not be a row of similar buttons for different functions, but a row of different buttons—for better, not in cows, but some arrangement contras- tively arranged. Do you need glasses to read what it says above the knobs? Loony. Can you not pull out a unit from another section? Good. CAN YOU WORK IT IN THE DARK?? Terrible.

As a rough guide, good design is inversely proportional to the probability of a user making a mistake, and this criteria carries over to interactive computer systems.

To make a system easy to use is extremely difficult and time consuming, in the same way that it takes more work to write a short article than a long one. You should not "design the system" first, and then put on a "friendly front" (although this is what must be done in many cases), any more than you should first shot a movie and decide what it is to be about (although this occasionally worked).

An interactive system should become second nature, and become second nature quickly. This is essential for many reasons. One is that we will have to deal with among many different interactive systems in the future, and there will be no time to acquire and adapt the local complications of each. They will have to spring clearly and straightforwardly at the mind and hand.

Moreover, interactive systems will be used intensively for hours, often by tired, high-strung, frenetic people, who will be expected to be getting a job done in a hurry, and who are thinking only of the world they are trying to operate in—not the intervening complications. It is up to us as designers to create fast, safe, elegant systems of view and operation without snags, dangers or complications.

The system designer, or movie director—he'll call him you must—has a full understanding of what things are easy to do, what things are not, and what is hopelessly impossible. You then make a collection of all the ideas and visualizations (and scripts and penas) that you would like to put together in your system. Then you make your rework them and rework them, and rework them.

THINK THE WORLD

—In many views and aspects (it is real nature (unlike what has been thought of as its nature).

IMAGINE ALL THE CONTROLS AND PRESENTATIONS YOU WOULD LIKE TO HAVE.

REDUCE THE CONTROLS AND PRESENTATIONS TO ANadequate, POWERFUL, EASY-TO-UNDERSTAND SET.

MARRY THEM TO THE AVAILABLE SCREENS, KEYS, AND POINTING TOOLS.

ABOVE ALL, DESIGN THE FULL SYSTEM FIRST—THEN CUT IT DOWN, IF YOU HAVE TO—you may FIND you DO NOT HAVE TO.

That is new thought is much more than the new design habit. Because unfortunately the salaried programmer has, in effect, a license to inflict on innocent users anything he likes under the pretext of technical necessity or on the basis of some off-the-cuff forOutline consultant's version of "user needs." I regard the decisions involved in design like those an intricate and non dependent as moves in chess. This kind of design needs a respect and even evidence for the unfurling ramifications of plans, and the staggering complexity of making things simple. I hope I have given a sense of this style of engine.

I hope, too, that the reader will see it as an art form—somewhere between moving, diagrams of machinery, the design of gaiters, and the building of philosophical systems.

When done well, it is done with simplicity, consistency, conceptual clarity and vividness. This is not "technical" work in any usual sense. I consider it a form of design and a form of art.

I believe that interactive design is more than anything else, what the computer field is really about. I find it intrinsically and appealing that these general principles are so little under- stood, that despite all the posh computer science curricula, nobody teaches these anywhere; and that innocent custo- mers who want an easy-to-use system, it is too much to ask?—are too often led by salesmen, and talks only the product path to col- lect horrors of complication instead of being prepared.

Most of all this is true when you spend hours designing what you think computers are all about and the world should be going.

If you were to write a letter to your family and friends—or financial backers—as a letter of clear exposition of complicated technicalities, then you don't want things to be easily comprehensible. If you were, you had to be making certain other verbal engagements.

But if you believe that somewhere beyond all the technicalities lies some kind of hope for a better future and a society well-managed, with ideas and knowledge and so on, you need to be talking about certain other personal engagements.

And you believe that somehow beyond all the technicalities lies some kind of hope for a better future and a society well-managed, with ideas and knowledge and so on, you need to be talking about certain other personal engagements.

And you believe that somehow beyond all the technicalities lies some kind of hope for a better future and a society well-managed, with ideas and knowledge and so on, you need to be talking about certain other personal engagements.

And you believe that somehow beyond all the technicalities lies some kind of hope for a better future and a society well-managed, with ideas and knowledge and so on, you need to be talking about certain other personal engagements.

And you believe that somehow beyond all the technicalities lies some kind of hope for a better future and a society well-managed, with ideas and knowledge and so on, you need to be talking about certain other personal engagements.