THE INFORMATION VISUALIZER, AN INFORMATION WORKSPACE

Stuart K. Card, George G. Robertson, Jock D. Mackinlay

Xerox Palo Alto Research Center
Palo Alto, California 94304
(415) 494-4362, Card.PARC@Xerox.COM

ABSTRACT

This paper proposes a concept for the user interface of information retrieval systems called an *information workspace.* The concept goes beyond the usual notion of an information retrieval system to encompass the cost structure of information from secondary storage to immediate use. As an implementation of the concept, the paper describes an experimental system, called the Information Visualizer, and its rationale. The system is based on (1) the use of 3D/Rooms for increasing the capacity of immediate storage available to the user, (2) the Cognitive Co-processor scheduler-based user interface interaction architecture for coupling the user to information agents, and (3) the use of information visualization for interacting with information structure.

KEYWORDS: Information retrieval, interface metaphors, information visualization, animation, desktop metaphor, UI theory, 3D graphics, interactive graphics.

INTRODUCTION

A new paradigm of computing use seems to be emerging in which computational aid will be applied to the storage, selection, and use of most sorts of information. Although data bases and information retrieval systems have been around for some time, the systems developed have relied largely on the power of search and indexing techniques. With some important exceptions (e.g., [1,9,11,12]) few systems have been noted for their user interfaces. Advances in computer technology have created new possibilities for information retrieval systems in which user interfaces could play a more central role. This paper proposes a paradigm for such interfaces--the *information workspace.* An implemented experimental system, the Information Visualizer, is developed as a specific instance of the paradigm.

THE COST STRUCTURE OF INFORMATION

Information retrieval has often been studied as if it were a self-contained problem (e.g., the library automation problem). Yet from the user's point of view, information retrieval is almost always part of some larger process of information use. What is really needed from the point of the user isn't so much information retrieval itself, but rather the amplification of information-based work processes (or other uses)--that is, methods and machines that would allow people to bring to bear on a task of interest more information more quickly than otherwise possible.

Consider, for example, an office worker as shown in Fig. 1. Information is available in the desk-side diary, through the computer terminal, in the immediate files on the desktop, through other people using the telephone, in books in the bookcase, in files in the filing cabinet. The sources of information take different forms--from paper documents to machines to people, but, nevertheless, each piece of information has a *cost* associated with finding and accessing it. Looked at abstractly, the office, at a particular moment, is characterized by a *cost structure* over the information in it. What is usually meant by an organized office is one with a cost structure arranged so as to lower the cost of the information-based work processes performed within it. File cabinets, desks, filing systems, and computer-based information retrieval systems can be thought of abstractly as just means for changing this cost structure of information.

The cost structure of the information in the office of Fig. 1 has been arranged with care (Its arrangement was derived from a brochure of a professional time management company [28]): A small amount of information (either frequently-needed or in immediate use) is kept where the cost of access is low--in an *Immediate Storage* area, principally the desk. Voluminous, less-used information is kept in a higher-cost, larger-capacity *Secondary Storage* area. More information is available in the library and other *Tertiary Storage* areas. In addition to these simplified categories, the information is linked and otherwise structured to aid in its retrieval.

In general, information processing systems, whether artificial, like this office, or natural, biological systems, like the human eye, tend to be organized to minimize the cost structure of information processing. General observations deriving from studies of these systems can help us to formulate systems goals for the design of user interfaces for information access. We consider six such observations.

> Observation 1 [HIERARCHY]. *Organizing the parts of a system hierarchically often improves the quantity of information processed relative to processing cost.* [22]

Hierarchical arrangement is one of the standard solutions used to achieve efficiencies in biological, socioeconomic, and engineering systems [26,22]. The eye is a familiar example of a information processing hierarchy. The office in Fig. 1 is an example of an information storage caching hierarchy.

> Observation 2 [HIGH COST RATIOS]. *The cost of accessing information often varies radically both because of the cost of finding it and because of the cost of assimilating it.*

The ratio between the cost of information ready to hand vs the cost of information not immediately available may be large, even orders of magnitude, as shown in Table 1. For example, in a typical computer system, the ratio between main memory speed and disk access time in a virtual memory system is 10^4~10^5. It may take a scholar months to discover and collect information materials relevant for a book. The juxtaposition of these materials, once they have been collected into a file or perhaps on a desktop, makes going back and forth among them relatively inexpensive.

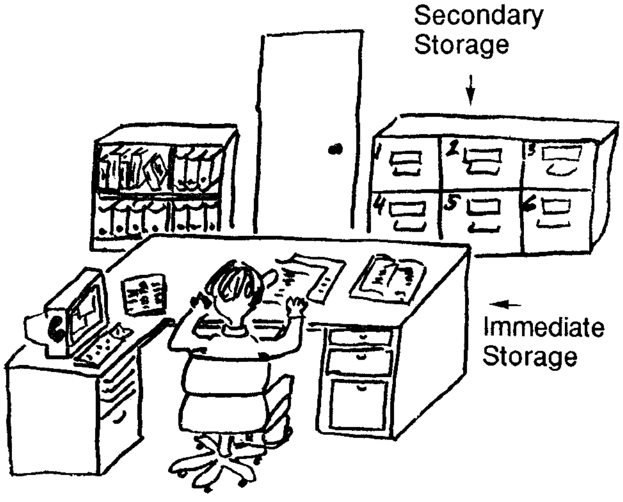


Fig. 1. An office organized to have an efficient information cost structure.

TABLE 1. COMPUTER MEMORY COST RATIOS

Storage type		Access time	Ratio
Immediate Storage	RAM	80 ns	
Secondary Storage	Disk	20 ms	2.5×10^5
Tertiary Storage	Optical	5 s	2.5×10^2

> Observation 3 [LOCALITY OF REFERENCE]. *The processing of information exhibits locality of reference. That is, over a small time interval, references to information are not uniformly distributed throughout the corpus, but tend to be concentrated in a subset, called the working set.*

This fact was first systematically encountered in studies of computer program memory use [8]. It also holds true if we look at the way in which people reference windows [6,14].

> Observation 4 [REFERENCE CLUSTERING]. *Information use defines clusters of information used repeatedly to perform some task. The processing of information tends to establish locality of reference in one cluster, then jump to another cluster. Some information may participate in more than one cluster.*

One might think that the information elements in a working set would gradually change as information processing proceeds. Instead, what actually tends to happen is that there is an abrupt transition to another working set of information elements [20].

> Observation 5. [MAX INFO/COST]. *Information systems tend to adjust themselves to maximize (or sometimes minimize) the quantify of information processed relative to some processing cost constraint.* [22]

An example of this observation is the way the visual system tends to encode points of greatest curvature in an image (these carry the most information [22]). Another example is the minimum work principle in conversation wherein the speaker attempts to anticipate some of the hearer's goals and reply with extra information that minimizes the joint cost of the exchange.

> Observation 6 [ABSTRACTION]. *Lower levels of an information processing system simplify and organize information, supplying higher centers with aggregated forms of information through abstraction and selective omission.* [22].

Even when information is in hand there is a generic problem that the volume of information to be processed is large relative to the abilities of the user. An answer is to stage processing by recoding the information in progressively more abstract and simpler representations.

The abstractions produced by the lower-level processing predetermine, to a considerable extent, the patterned structures that the higher-level processing can detect [22]. The higher-level processing, in turn, reduces still further the quantity of information by processing it into yet more abstract and universal forms. In biological systems, this process allows the mixing of information generated through different sensory modalities.

INFORMATION WORKSPACES

If we want to move beyond information retrieval, narrowly conceived, to address the amplification of information-based work processes, we are led to try to develop user interface paradigms oriented toward managing the cost structure of information-based work. This, in turn, leads us to be concerned not just with the retrieval of information from a distant source, but also with the accessing of that information once it is retrieved and in use. And this problem, the necessity of lowering the cost of work by providing some sort of low-cost, immediate storage for accessing objects in use is a common problem faced by most kinds of work. The common solution is a *workspace*, whether it be a woodworking shop, a laboratory, or an office. A workspace is a special environment in which the cost structure of the needed materials is tuned to the requirements of the work process using them.

Computer screens also provide a workspace of a sort for tasks done with the computer (and, of course, may be components of larger workspaces, such as offices). Computer-screens as workspaces have gone through several stages of evolution (Fig. 2). Early workspaces were command-based scrolling teletypes or their CRT equivalent and this style survives in DOS and UNIX systems today. Engelbart's NLS system [10] introduced the notion of direct interaction with the text in documents using a point and click editor based on the mouse.

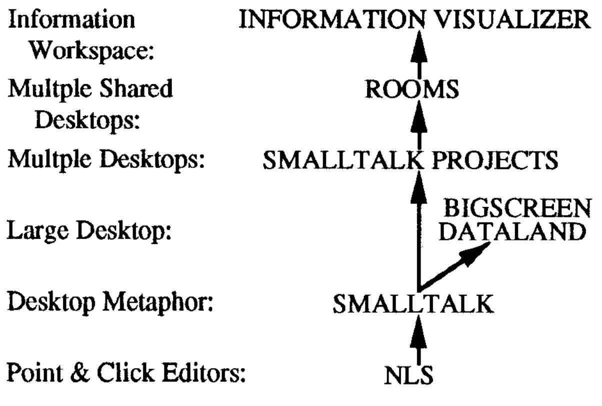


Fig. 2. Evolution of computer workspaces.

The next stage was the desktop metaphor developed as part of the Smalltalk [15] and Star [27] systems. By adding menus, windows, and icons to mouse-based direct interaction, these systems allowed the workspace to contain multiple documents and shifted user processing from recall-based to recognition-based. The desktop metaphor affected the cost structure of information by allowing low cost access to more information in the Immediate Storage environment. Smalltalk Projects introduced the notion of multiple workspaces that users could switch among, allowing still more information to reside in the immediate work area (but at the added cost of switching and finding the right workspace).

A potential next step was the large, extended desktop metaphor, introduced by Dataland [1]. The problem with the single large workspace (as we found in BigScreen [14], another attempt at a large desktop) is that the cost of search for relevant parts of the workspace rapidly increases with the number of elements in the workspace (unless the space itself has meaning as in a city map or a grocery store). Even the CCA version [1] of the Dataland system, developed for practical use, abandoned the single large workspace in favor of multiple workspaces.

The Rooms system [4,14] added to the multiple desktop notion an ability to share the same information objects in different workspaces, both individually and as groups. It also added an overview and other navigational aids as well as an ability to store and retrieve workspaces. These removed the major disadvantages of multiple desktops.

The essence of our proposal for a user interface for information retrieval is to evolve the Rooms multiple desktop metaphor into a workspace for information access--an *information workspace.* Unlike the conventional information retrieval notion of simple access of information from some distal storage, an information workspace (1) treats the complete cost structure of information, integrating information access from distant, secondary or tertiary storage with information access from Immediate Storage for information in use, and (2) considers information access part of a larger work processes. That is, instead of concentrating narrowly on the control of a search engine, the goal is to improve the cost structure of information access for user work. Our intention is to build on progress in information retrieval studies, but to do so by reframing the problem as the amplification of information-dependent work processes.

The Information Visualizer described in this paper is our experimental embodiment of this concept. The Information Visualizer has three major components (see Table 2): (1) *3D/Rooms*, a 3-D version of the Rooms system. This component effectively increases the capacity of Immediate Storage, thus making a more effective hierarchical storage system in accord with Observations 1-4. (2) The *Cognitive Co-Processor* [23], an animation-oriented user interface architecture. This component increases the rate of user-system interaction and information transfer in accord with Observation 5. And (3) *information visualizations*, which serve as structure-oriented browsers into sets of information. This component increases the level of abstraction

TABLE 2. SUMMARY OF THE DESIGN RATIONALE FOR THE INFORMATION VISUALIZER.

PROBLEM	ANALYSIS/ OBSERVATIONS	SYSTEM GOALS	UI ARTIFACT		
			3D/ROOMS	COGNITIVE CO-PROCESSOR	INFORMATION VISUALIZATIONS
Aid information access& processing	1. HIERARCHY 2. HIGH COST RATIOS 3. LOCALITY OF REF. 4. REF. CLUSTERING	More Immediate Storage			
		Larger	●	○	○
		Denser	●	●	●
		Cheaper Secondary Storage access	○	○	●
	5. MAX INFO/COST	Highly-coupled systems			
		Iterative retrieval			
		Faster cycle	○	●	○
		Fewer cycles	○	○	●
		Cognitive impendance match	○	●	○
	6. ABSTRACTION	Information visualization			
		Linear structure	○	○	●
		Hierarchical structure	○	○	●
		Continuous data	○	○	●
		Geographical data	○	○	●

abstraction between the user and the available information in accord with Observation 6.

Increasing Immediate Storage: 3D/Rooms

Let us return to the information cost structure of the office in Fig. 1. Clearly, it is advantageous to put more information into the cheaper Immediate Storage area. The problem (for a desktop or a computer desktop metaphor) is that the storage only holds so much. When the storage is overloaded by overlapping information, then searching time can raise the access cost of Immediate Storage radically and non-linearly, defeating the purpose of Immediate Storage and leading to phenomena analogous to thrashing [14].

Observation 3, locality of reference, shows that a local workspace is viable (and this is why the desktop metaphor works). Observation 4, reference clustering, tells us that the working set of items referenced is likely to undergo sudden shifts to other working sets. This is the basis of the Rooms multiple workspace concept, which allows the user to organize his or her work into (partially overlapping) sets of items and to switch among them easily. Rooms effectively makes the Immediate Storage *larger* by having multiple desktops (at some cost increase due to switching). It avoids the overload problem, by switching the screen to show only those items in the active Room.

3D/Rooms extends this logic to a 3D workspace. In the classical desktop metaphor and the original Rooms system, the view of a Room is fixed. In 3D/Rooms, the user is given a position and orientation in the Room. He or she can move about the Room, zoom in to examine objects closely, look behind himself, or even walk through doors into other Rooms. The 3D/Rooms workspace is sufficiently different from the desktop metaphor that it requires a new set of building blocks: We developed a walking metaphor algorithm for exploratory movement of the user [17] and a point of interest logarithmic movement algorithm for very rapid, but precise movement relative to objects of interest [18]. Another algorithm allows 3D objects to be moved rapidly using only the mouse [18]. 3D/Rooms is built upon an animation-based user interface architecture--the Cognitive Co-processor [23]. The buttons of Rooms have been developed into autonomous interactive objects. Like Rooms, 3D/Rooms contains an overview allowing the user to view all the 3D workspaces simultaneously (Card Plate 1). But in 3D/Rooms the user can actually reach into the Rooms from the overview, move about in them, and manipulate their objects.

The effect of 3D/Rooms (and the associated Cognitive Co-processor architecture) is to make the Immediate Storage not only larger, but also *denser*. J. J. Gibson has made the point that perception is an active process [13]. Instead of identifying what is visible on the screen with the user's perceptual awareness, it is probably more accurate to think of the visual and perceptual system as a flying spot scanner that is updating an internal awareness [22]. Just as a driver is aware of a car behind him glimpsed in his mirror a few seconds previous, the user seems to be able to be aware of objects that are "behind" him or occluded if he can easily

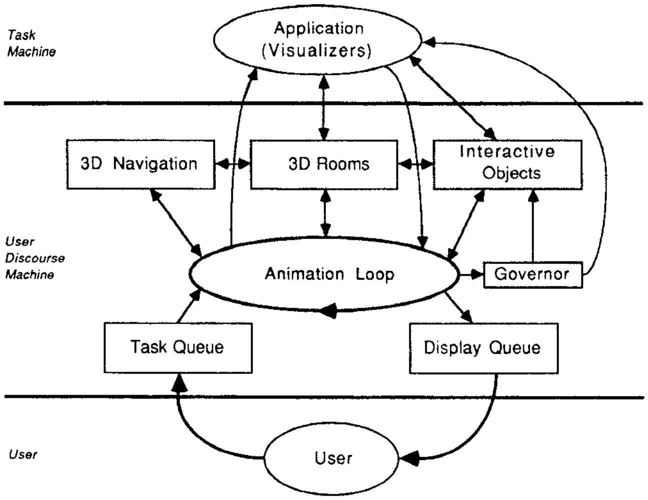


Fig. 3. Cognitive Co-processor interaction architecture.

(say within about a second) update his knowledge of them. Thus interactive animation and 3D perspective graphics both allow us to apply Gibson's active perception tenants and to pack the space more densely with information than would otherwise be possible. By manipulating objects or moving in space, the user can disambiguate images, reveal hidden information, or zoom in for detail--rapidly accessing more information. For example, in a companion paper [24], we describe a corporate organization tree requiring 80 pages on paper that has been displayed in a single 3D/Rooms screen.

Increasing Information Interaction Coupling with the User: the Cognitive Co-processor

Except in special circumstances, it is difficult for a user to ask an information retrieval system what he or she wants, because the user does not, in general, know what is available and does not know from what it has to be differentiated. For this reason, we have adopted an iterative retrieval philosophy, in which the user is assumed to iterate with the system through several cycles until the use finds what is wanted. This is, in fact, similar to what people do when they make requests to reference librarians. In order to maximize the information gained as a function of time cost, we attempt, on the one hand, to increase the speed of this iterative cycle as much as possible (for example, by using graphics to speed user assimilation), and on the other hand to decrease the number of times necessary to go around the cycle (for example, by using fish-eye [12] focus + context displays to speed navigation).

To support rapid cycles of the user-system interaction, we have created a new user interface interaction manager substrate called the Cognitive Co-processor, derived from Sheridan's notion of supervisory control [25,2]. The idea is that the user is trying to control possibly multiple applications running as semi-autonomous agents. The Cognitive Co-processor (Fig. 3) is meant to be a sort of "impedance matcher" between the cognitive and perceptual information processing requirements of the user and the properties of these agents. For example, in general these agents operate on time constants different from those of the user. There are three sorts of time constants for the human that we want to tune the system to meet: perceptual processing, immediate response, and unit task (Table 3).

TABLE 3. HUMAN TIME CONSTANTS FOR TUNING COGNITIVE CO-PROCESSOR

TIME CONSTANT	VALUE	REFERENCES
Perceptual processing	.1 s	[5]
Immediate response	1 s	[21]
Unit task	10 s	[5,21]

The perceptual processing time constant. The Cognitive Co-processor is based on a continuously-running scheduler loop and double-buffered graphics. In order to maintain the illusion of animation in the world, the screen must be repainted at least every .1 sec [5]. The Cognitive Co-processor therefore has a *Governor* mechanism that monitors the basic cycle time. When the cycle time becomes too high, cooperating rendering processes reduce the quality of rendering (e.g., leaving off most of the text during motion) so that the cycle speed is increased.

The immediate response time constant. A person can make an unprepared response to some stimulus within about a second [21]. If there is more than a second, then either the listening party makes a backchannel response to indicate that he his listening (e.g., "uh-huh") or the speaking party makes a response (e.g., "uh...") to indicate he is still thinking of the next speech. These serve to keep the parties of the interaction informed that they are still engaged in an interaction. In the Cognitive Co-processor, we attempt to have agents provide status feedback at intervals no longer than this constant. Immediate response animations (e.g., swinging the branches of a 3D tree into view) are designed to take about a second. If the time were much shorter, then the user would lose object constancy and would have to reorient himself. If they were much longer, then the user would get bored waiting for the response.

The unit task time constant. Finally, we seek to make it possible for the user to complete some elementary task act within 10 sec (say, 5~30 sec) [5,21], about the pacing of a point and click editor. Information agents may require considerable time to complete some complicated request, but the user, in this paradigm, always stays active. He or she can begin the next request as soon as sufficient information has developed from the last or even in parallel with it.

Increasing the Abstraction of Information--Information Visualization

According to Observation 6, information processing often proceeds by lower levels in the system simplifying information through aggregation, abstraction and selective omission. In this way, large amounts of raw information are reduced to volumes within the capacity of the higher centers and the abstractions introduced by the lower centers can be further aggregated into patterns.

Recent work in scientific visualization shows how the computer can serve as an intermediary in the process of abstraction. Large sets of data are reduced to graphic form in such a way that human perception can detect patterns revealing underlying structure in the data more readily than by a direct analysis of the numbers. Information in the form of documents also has structure. *Information visualization* attempts to display structural relationships and context that would be more difficult to detect by individual retrieval requests.

Some abstract structures are common across many information sets. One example is *hierarchical structure* (e.g., UNIX directories). A companion paper [24] describes structural browsers called the Cone Tree and the Cam Tree, based on an animated 3D visualization of hierarchy. Another example is *linear structure*. We discovered in field observations of an architect's office, for example, that time of creation was one of the most important retrieval attributes of a document since it related so intimately to the work process. In another companion paper [19], we describe a structural browser, called the Perspective Wall that allows retrieval using a visualization of linear structures.

A third example is a 2D continuous data surface similar to much scientific data. In this case our visualization is a Data Sculpture (see Card Plate 2). The user can walk around or zoom into this visualization containing over 65000 sampling points as if it were a sculpture in a museum. The user can also manipulate some of its viewing parameters. A fourth example is the spatial structure of a building. Card Plate 3 shows a portion of an office building used as a structural browser for people. Selecting an organization will produce the names and pictures of its members and select their offices. Clicking on offices retrieves their inhabitants.

These visualizations use interactive computer graphics animation to explore dynamically changing views of the information structures. More visualizations are visible in the Rooms overview of Card Plate 1. The visualizers attempt to present abstractions of large amount of data tuned to the pattern detection properties of the human perceptual system. For example, they use color, lighting, shadow, transparency, hidden surface occlusion, continuous transformation, and motion cues to induce object constancy and 3D perspective

Indexing and Searching

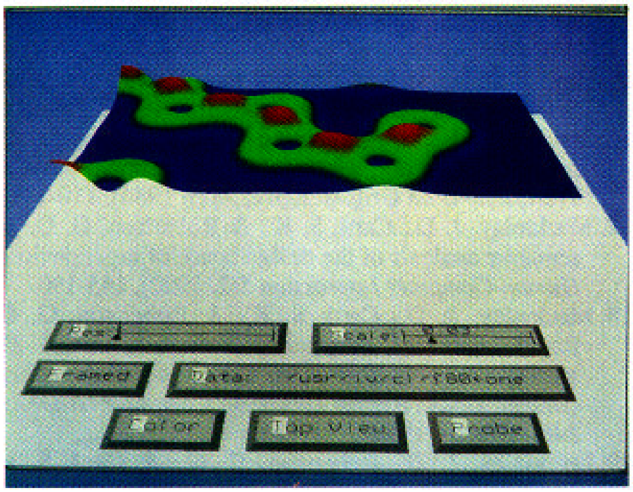
We have concentrated here on the user interface paradigm aspects of the Information Visualizer. The Information Visualizer is based on an indexing and search subsystem built by other members of our group [7]. Briefly, this subsystem TDB provides stemming and a full text inverted database. Documents are represented as word vectors as are requests. This allows us to search for documents given a set of descriptors or to use documents themselves as the retrieval request to find other documents in an iterative "relevance feedback" retrieval paradigm. In one demonstration, for example, biographies of several staff members who are linguists are selected. The result is that on the next retrieval iteration more linguists are retrieved. Associative retrieval based on such linguistic searches can be used to highlight portions of the information visualization. Thus we can combine associative and structural searches.

DISCUSSION

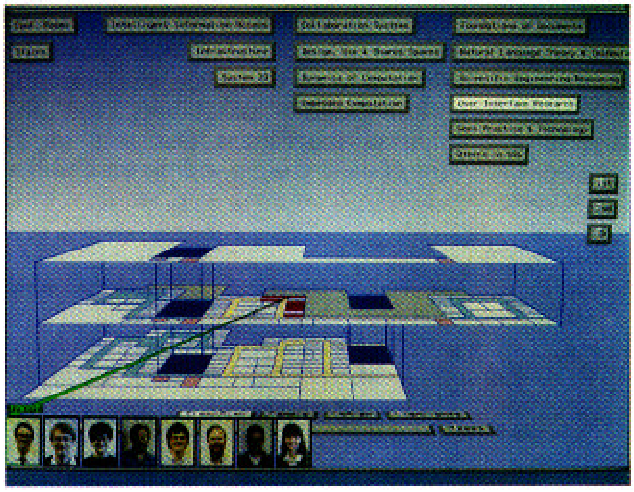
The Information Visualizer we have described is an experimental system being used to develop a new user interface paradigm for information retrieval, one oriented toward the amplification of information-based work. It is based on our analysis of several aspects of information use that have led us to reframe the information retrieval problem as a problem in the cost structuring of an information workspace. This, in turn, has led us to evolve the computer desktop metaphor toward (1) 3D/Rooms (to manage information storage cost hierarchies), (2) the Cognitive Co-processor interaction architecture (to support highly-coupled iterative interaction with multiple agents), and (3) information visualization (to increase the level of information abstraction to the user).

This paper continues a set of studies in which we have attempted to integrate theoretical and empirical analyses with practical system design, either through the analytical characterization of existing designs [5,16,17] or the synthetic generation of new designs based on analytical underpinnings [4,14,16,18,23]. The development of theoretical methods and practical designs in engineering-oriented disciplines tends to take a different course than in the natural sciences owing to the particular interplay of synthetic and analytic activities. This course can be summarized as the *systems research paradigm*: (1) Initial exploratory point designs demonstrate the possibility of obtaining some performance. These may be incrementally improved through cut and try. (2) Abstractions are developed describing the structure of regions in the design space. (3) Theories and empirical observations are used to characterize sub-regions of the design space, showing which designs lead to what performance. (4) Finally, knowledge of the design space is codified in such a way that it can be transmitted as a body of knowledge to other people who need to build such systems.

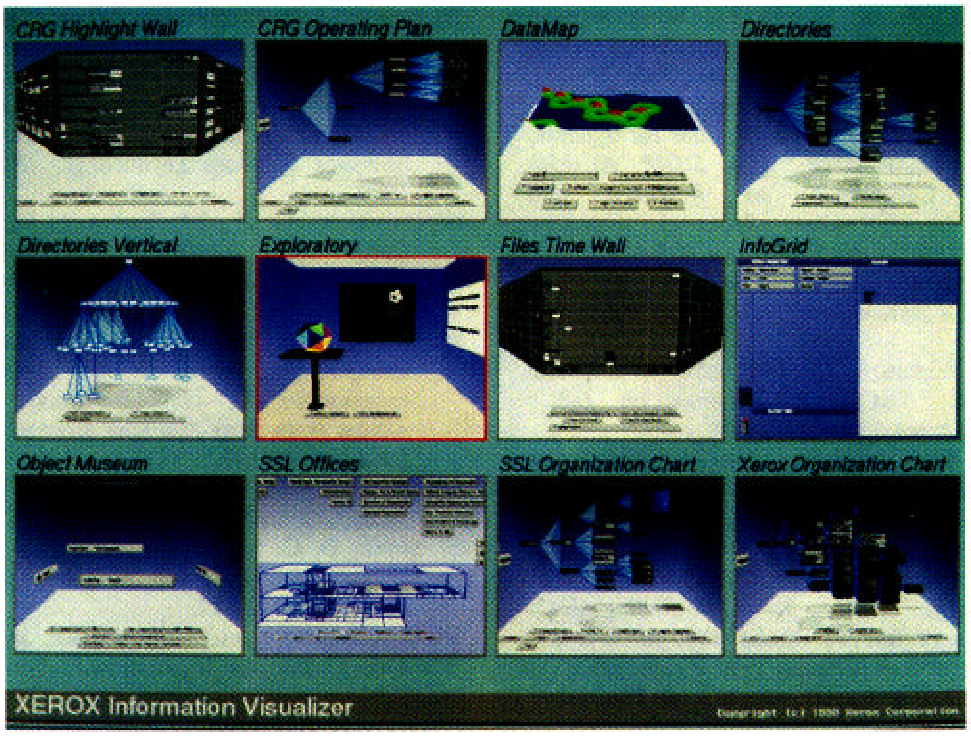
This sequence reverses the more familiar natural science course of theory to application. But, actually, the interplay of system synthesis, abstraction, and analysis may play out in almost any order, and there can be subregions of the design space in very different states. In the present study we have been able to utilize theory and empirical relationships established from previous research [5,6,8,12,14,20,21,22,25,26] as well general theoretical observations from the literature as "tools for thought" to



Card Plate 1



Card Plate 2



Card Plate 3

conceptualize the problem, even though we have adopted an exploratory system-building methodology. The exploratory system building, in turn, brings us to a position where we will be able to perform empirical use studies and design characterization. Regardless of the order, the general need is both for new user interface paradigms that utilize emerging technological possibilities and the analytical and empirical foundations that help us to understand the merits of these designs and the possibilities for new ones.

REFERENCES

1. Bolt, R. A. *The Human Interface.* Lifetime Learning Publications, Belmont, California, 1984.
2. Card, S. K. Human factors and artificial intelligence. In P. A. Hancock & M. H. Chignell (Eds.), *Intelligent Interfaces: Theory, Research and Design.* Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1989.
3. Card, S. K. Information Workspaces. *Friend 21 Conference* (Tokyo, September), 1989.
4. Card, S. K. & Henderson, A. H, Jr. A multiple, virtual-workspace interface to support user task switching. *Proceedings of the CHI+GI 1987* (Toronto, April 5-7). ACM, New York, 1987, pp. 53-59.
5. Card, S. K., Moran, T. P., & Newell, A. *The Psychology of Human-Computer Interaction.* Erlbaum, Hillsdale, New Jersey, 1983.
6. Card, S.; Pavel, M., & Farrell, J. Window-based computer dialogues. In Brian Shackel (Ed), *Human-Computer Interaction--Interact '84* (September 4-7). North-Holland, Amsterdam, 1984, pp. 239-243.
7. Cutting, D. & Pedersen, J. Optimizations for dynamic inverted index maintenance. *ACM SIGIR '90, 13th International Conference on Research and Development in Information Retrieval.* (Brussels, September 5-7), ACM, New York, pp. 405-411.
8. Denning, P. J. *Working sets past and present. IEEE Transactions on Software Engineering, SE-6*(1 1980): 66-84.
9. Donahue, J. & Widom, J. Whiteboards: A graphical database tool. *ACM Transactions on Office Information Systems 4*(1 1986), 24-41.
10. Engelbart, D. C. *Augmenting human intellect: A conceptual framework* (AFOSR-3223). Stanford Research Institute, Menlo Park, California, 1962.
11. Fairchild, K. M.; Poltrock, S. E.; & Furnas, G. W. SemNet: Three-dimensional graphic representation of large knowledge bases. In R. Guindon (Ed.), *Cognitive Science and its Applications for Human-Computer Interaction.* Erlbaum, New Jersey, 1988.
12. Furnas, G. Generalized fisheye views. In M. Mantei and P. Orbeton (Eds.), *ACM CHI '86 Conference on Human Factors in Computing Systems* (Boston, Massachusetts, April, 14-18). ACM SIGCHI, New York, 1986, 16-23.
13. Gibson, J. J. *The Ecological Approach to Visual Perception.* Boston: Houghton Mifflin, 1979.
14. Henderson, D. A., Jr. & Card, S. K. Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics 5*(3, July 1986), 211-243.
15. Kay, A. Microelectronics and the personal computer. *Scientific American 237*(3, September 1977), 230-244.
16. Mackinlay, J. D. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics, 5*(2, April 1986), 110-141.
17. Mackinlay, J. D., Card, S. K., & Robertson, G. G. A semantic analysis of the design space of input devices. *Human-Computer Interaction 5*(2, 1990), 145-190.
18. Mackinlay, J. D., Card, S. K., & Robertson, G. G. Rapid controlled movement through a virtual 3D workspace. *SIGGRAPH '90 Conference Proceedings* (Dallas, Texas, August 1990). In *Computer Graphics, 24*(4, August, 1990), 171-176.
19. Mackinlay, J. D., Robertson, G. G., & Card, S. K. Perspective wall: Detail and context smoothly integrated. *ACM CHI '91 Conference on Human Factors in Computing Systems* (New Orleans, April), ACM, New York, 1991.
20. Madison, A. W. *Characteristics of Program Localities.* Ann Arbor, Michigan: University Microfilms International, 1992.
21. Newell, A. *Unified Theories of Cognition.* Harvard University Press, Cambridge, Massachusetts, 1990.
22. Resnikoff, H. L. *The Illusion of Reality.* New York: Springer-Verlag, 1989.
23. Robertson, G. G., Card, S. K., Mackinlay, J. D. The cognitive co-processor architecture for interactive user interfaces. *ACM SIGGRAPH Conference on User Interface Software Technology.* ACM, New York, 1989.
24. Robertson, G. G., Mackinlay, J. D., & Card, S. K. Cone Trees: Animated 3D visualizations of hierarchical information. *ACM CHI '91 Conference on Human Factors in Computing Systems*, 1991.
25. Sheridan, T. B. Supervisory control of remote manipulators, vehicles, and dynamic processes: experiments in command and display aiding. *Advances in Man-Machine Research, 1*, JAI Press, 1984, pp. 49-137
26. Simon, H. A. *The Sciences of the Artificial.* MIT Press, Cambridge, Mass, 1969.
27. Smith, D. C. Irby, C., Kimball, R. Verplank, W. & Harslem, E. *Designing the Star user interface. Byte* 7(4, 1982), 242,282.
28. Time Manager International. *Me, My Time, My Life.* Time Manager International, Hillerod, Denmark, 1981.