



Tools and Techniques for Creative Design

“A WEEKDAY EDITION OF *THE NEW YORK TIMES* contains more information than the average person was likely to come across in a lifetime in seventeenth century England” [Wurman, 1989]. The world is rich with data, rapidly becoming more varied in media type—sounds, photos, FAX, computer-generated images, telephony, as well as text flood our lives. The challenge is how to best present and represent such data within the interface, to transform it into useful information.

For computer users, much so-called information is virtually inaccessible, not only because there is so much of it, but also because the many types of information formats and delivery systems are not integrated at the interface. Electronic mail, for instance, comes in a bewildering assortment of flavors: screen text, voice, or paper print-out; usenet, MCI, or interoffice e-mail; personal messages, electronic conferences, or bulletin boards. Telecommunications, service industries, entertainment companies, and educational institutions will use tomorrow’s personal information environments to store and present enormous worlds of data. Several business areas do not yet provide extensive computer access to their workers; for example, theatrical production, fast-food chains, creative art and design companies, travel-related industries, bookings, and merchandise ordering in retail sales. Future interfaces need to incorporate new information types and to accommodate new types of users with additional customized real-world interface metaphors that make information easy to find and use.

S. Joy Mountford

*Human Interface Group
Apple Computer, Inc.*

With the onslaught of more varied information and application types, there is a great need for guidance in generating more creative, usable interface designs. We need techniques to help us do a better job of visualizing the look and feel of potential interfaces. Such interface prototyping tools as HyperCard have been a great step forward in enabling graphic designers to demonstrate their ideas, but we need even more powerful tools with which to emulate the “feel” of the interface in the hands of new users with new uses. A limited number of computer tools are available to assist in the design process (see the chapter by Wagner in this volume), but even fewer exist to help us decide which ideas are worthwhile.

The science and art of computer interface design is a relatively young discipline within a world of more mature design traditions. There are few success stories for interface designers to learn from, compared with traditional disciplines such as mathematics and biology. There are even fewer principles that can help us predict the behavior of our clients, the users. We have much to learn about interpersonal communication, group behavior, aesthetics, showmanship, and users’ adaptability to change. Some of our new interface ideas will come from people who study thought, language, entertainment, and communication, as well as from people who study hardware, algorithms, data structures. We need to find better mechanisms for facilitating collaboration among designers from both artistic and scientific points of view, as articulated in “The Role of the Artist in the Laboratory” [Buxton, 1988].

This paper examines the emerging discipline of interface design in terms of established disciplines, the present state of the art, and techniques for generating new ideas. Looking to the past, we can identify mature design disciplines in other fields that can inform our own. The first section of this paper examines the areas of “traditional” animation, theater, architecture, industrial design, and information display for techniques and approaches that can be of use to human-computer interface designers. The second section looks at the more contemporary tools of metaphor and user observation. The final section of the paper surveys some techniques for designing the interfaces of the future through brainstorming and the creative juxtaposition of ideas.

New Wine from Old Vats

FILM

Table 1 shows an insightful historical timeline indicating some of the changes that have occurred to various communication crafts as they moved from being inventions to being art forms (this table is taken from *The Elements of Friendly Software Design* [Heckel, 1984]). Heckel views “software as a newcomer to communication crafts.” It is interesting to note that one of the most recent “arts” to change its form is filmmaking, which

Writing	Painting	Film	Drama	Photography	Architecture	Music	Computers
3300 BC Egyptian hieroglyphics	20000 BC Cave Paintings of Lascaux, France	1884 Thomas Edison demonstrates kinetoscope	600 BC First Greek amphitheatre built for festival of Dionysos	1839 Daguerre announces his photographic method	2700 BC Pyramids at Sakkara and Gizeh	2000 BC Invention of the Egyptian harp	450 BC First abacus
2300 BC Maxims of Ptohotep	3000 BC Egyptian tomb paintings	1903 Edwin Porter's <i>The Great Train Robbery</i>	550 BC Introduction of the chorus in Greek plays	1861 Mathew Brady's civil war photographs	400 BC Parthenon Athens	1306 AD First pedal organ	1864 Charles Babbage's analytic engine
1500 BC Development of phonetic alphabet	1100 AD Invention of oil paints	1915 D.W. Griffith's <i>Birth of a Nation</i>	442 BC Sophocles' <i>Antigone</i>	1882 Eastman develops Kodak camera	1100 AD Development of flying buttress	1404 First clavichord	1955 First commercial mainframe computer
800 BC Library of Assyria founded	1497 Da Vinci's <i>Last Supper</i>	1927 First Sound Movie: <i>The Jazz Singer</i>	1580 AD Establishment of the first repertory companies	1902 Stieglitz's "Spring Showers"	1400 Completion of Chartres Cathedral	1520 First violin	1958 Fortran: First high level language
1440 AD Gutenberg's invention of moveable type	1654 Rembrandt's <i>Portrait of Jan Six</i>	1941 Orson Welles' <i>Citizen Kane</i>	1603 Shakespeare's <i>Hamlet</i>	1947 Ansel Adams' Yosemite photographs	1920 Invention of pre-stressed concrete	1602 Monteverdi's first opera, <i>Orfeo</i>	1975 First Personal Computer
1605 Cervantes' <i>Don Quixote</i>	1937 Picasso's <i>Guernica</i>				1936 Frank Lloyd Wright's "Falling Water"	1720 J.S. Bach: <i>Well-Tempered Clavier</i>	1978 VisiCalc
1719 First Modern Novel: Defoe's <i>Robinson Crusoe</i>						1824 Beethoven's Ninth (Choral) Symphony	1979 WordStar

Table 1: Most communications crafts started as inventions and evolved slowly in the direction of an art form. Software is a newcomer to the world of communications crafts. Reprinted by permission of Warner Books, Inc., from *Elements of Friendly Software Design*, copyright © 1984 by Paul Heckel.

likewise had its roots as an engineering discipline. Originally movies were viewed as having a fixed window within which some action occurred; they were often nothing more than continuous slideshows of information. The roots of filmmaking changed with the production of *Birth of a Nation* by D. W. Griffith in 1914. Engineers who had controlled films actually lost out to those who saw the artistic possibilities for film. Griffith's success began the relatively recent tradition of films as an art form, communicating directly to an audience.

The next "new" communications medium is likely to be that of software design. Like film, software was originally dominated by those who knew how to program, but this has now expanded to include a much larger audience. The first most widely used successful piece of software was VisiCalc™, which was considered "friendly" and usable by a wide audience who had hitherto found financial packages rather cumbersome. Tools such as HyperCard™ have made the price of entry into so-called "program-

ming,” or rather scripting, much more accessible to a range of new, more artistic, skill sets. It would serve interface designers well to learn from the lessons of filmmaking, to facilitate the best growth directions for interface design. Our new interfaces will serve as future vehicles of communication to a large audience, the world’s users.

ANIMATION

Walt Disney Studios produced their most famous animation classics long before computer animation existed. Disney’s goal was to create an animated form of communication that was as new in the 1920s as computers were in the 1960s. Both are powerful communicative forms that rely heavily on the use of graphics. We employ limited animation in the Macintosh interface today—the zooming effect for opening and closing of files, for instance. New uses for animation in human-computer interfaces are explored in the chapter by Baecker and Small in this volume. Tomorrow’s personal computers have the potential to create animations rich enough to allow users to actually interact with dramatic dynamic events. The principles developed by the Disney animators, who worked without automated assistants, are likely to prove quite useful in interface design.

A set of twelve principles of animation is detailed in *Disney Animation: The Illusion of Life*, by Thomas and Johnston [1984]. Anticipation, for example, is a motion characteristic that can both provide information and produce powerful dramatic effects. Disney’s animated characters exaggerate the way bodies prepare to move forward by pulling backwards slightly beforehand. Anticipation “telegraphs” the character’s intent, adding visual continuity and a sense of realism that carries the audience forward with the character. Disney also used sound quite effectively to represent such aspects of events as position, direction, and speed of motion. Sound can be similarly applied to interface events to establish continuity and provide information for the user (see the chapter by Mountford and Gaver in this volume). Each of Disney’s principles of animation contains valuable insights and techniques for interface designers.

THEATER

The dramatic arts are full of promising ideas for interface design. Contemporary discussions of software agents with humanlike qualities brings one to consider exactly what makes characters effective and believable (see the chapter by Laurel in this volume). A recent research project at Apple called Guides demonstrated a way of searching a database with the help of different personal viewpoints. These personal guides gave users their individual viewpoints of a encyclopedic slice of Grolier’s American history. Initial studies indicated that users found the guides engaging and informative (see the chapter by Oren et al. in this volume). Dramatic stories and

characters are familiar and effective mechanisms for structuring thought and behavior. We have barely begun to scratch the surface in applying this body of knowledge to computer interface design.

Drama has the power to *engage* audience members both emotionally and cognitively. Interactive plays and novels are beginning to be studied to see what facets make them more or less effective with readers and viewers. By isolating the key factors contributing to audience involvement and participation, we can discover techniques for enhancing direct engagement through human-computer interfaces [Laurel, 1986].

ARCHITECTURE

Architecture and interface design have an important goal in common: to create livable, workable, attractive environments. The principle of "form follows function" maintains that the form of objects should follow their functional requirements. The architectural movement towards functionalism was initially contested in Victorian England, as the industrial age was trying to hide the impact of industrialization on people's everyday lives. Le Corbusier was initially viewed with skepticism when he proposed that a house was a *machine for living*, as opposed to the traditional idea that a house was a shelter. He likened a house to a ship or bridge in its function. Likewise, we are only just beginning to conceive of computers as extensions of our functional everyday lives.

The listing of all the requirements to be met on an architect's design project have become known as *programming*. The more precise these definitions can be made on a project, the less likely it is that variations will occur among the different contributors to any design project. In architecture, these requirements are often referred to as *constraints*, just as they are known to interface designers. In order to make the process of designing within constraints as systematic as possible for architects, Alexander [1964] devised a set of steps for listing all functional requirements for a given architectural design problem. As an example, he specifies 21 requirements for designing a tea kettle. He then studies the interactions between the requirements and groups them into related clusters that form a hierarchical order of relationships that will govern the final design.

Managing trade-offs becomes rather difficult when the architect cannot control all of the variables. For example, requirements for plenty of natural light and interior quiet may be in conflict. If the windows must face a noisy street for the best light, then a trade-off decision must be made. Such trade-off decisions can be made in isolation by even an inexperienced designer, but experienced designers can often generate better alternatives because they have a better feel for the relative importance of the various requirements. Experience can lead to more graceful solutions. As Pile observes:

Satisfactory performance is invariably a matter of meeting a number of requirements—a vast number in the case of complex designs. The various requirements may be in conflict so that different satisfactory solutions can result from differing emphases on differing requirements. One most satisfactory solution only can emerge in the limited context of one particular evaluation of the relative importance of various desiderata. [Pile, 1979]

Alexander captures some of the wisdom of an experienced designer in his formal methodology. It can improve the performance of the inexperienced, and can help even an experienced designer make a more systematic analysis of the consequences of alternative solutions. This approach is controversial because it says little about beauty, harmony, or pleasure. Even so, it can be a spur to creativity in the fields of both architecture and interface designs.

INDUSTRIAL DESIGN

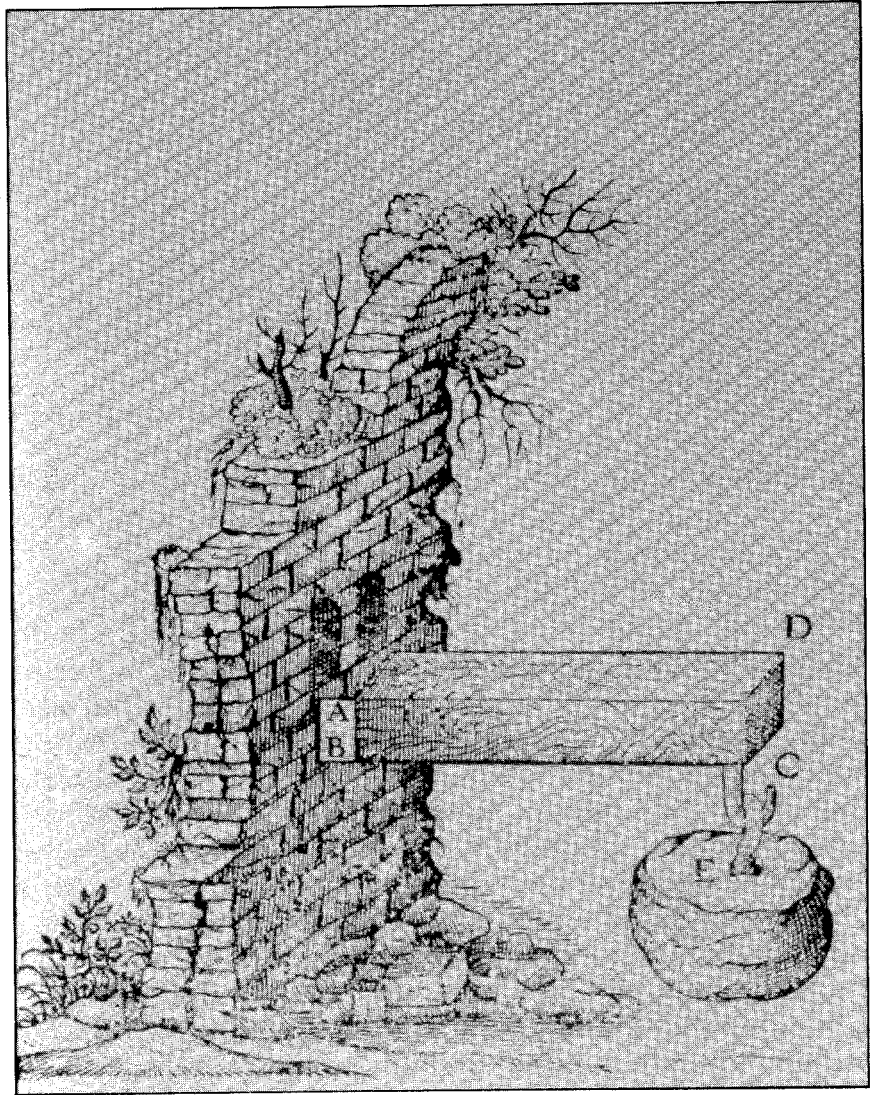
The experience of designing real-world objects for everyday use has a lot to teach us about how to design usable interfaces for software. Those designs which endure the test of time are mostly those that are reliable, offering straightforward ease of use and pleasurable user experiences. The chapter by Vertelney and Booker in this volume suggests principles and techniques of industrial design that can guide our interface work.

INFORMATION DISPLAY

In applications ranging from statistical graphics to scientific visualization, computers are used increasingly as tools for creating graphical displays of information. Often, however, the designers of such displays fail to take advantage of the techniques developed in “traditional” information display disciplines. The graphical prowess of today’s computers has encouraged the production of a plethora of unreadable graphics, characterized by Tufte as “chartjunk” [see Verity, 1985]. Tufte goes on to observe that “people who don’t see particularly well are designing interfaces to graphics machines. People have been doing books for 500 years and much wisdom has come from all that activity” [Tufte, 1983]. Not just interface designers, but American culture as a whole, seems to undervalue the established disciplines in this area—in Japan, for instance, there is an Annual Statistics Day contest and celebration! Experts in traditional fields like statistics and cartography can provide wisdom and techniques to the designer of graphical information displays (see figure 1).

The domain of classical mathematics employs a linear language of symbolic expression that is poorly suited to the representation of parallel events and processes, yet this is often the first “language” adopted by interface designers—the software for creating 3-D graphics on a leading graphics workstation, for instance, requires the user to define figures and specify

Figure 2: A sketch from Galileo's analysis of the strength of a beam. The sketch obscures its information content with a welter of irrelevant detail. Obtained from the Ann Ronan Picture Library, Somerset, England.



automatically. A map, for example, provides specific locational information using grid coordinates, which for some people is more useful than analogic spatial arrangement. Examples of frequently used perceptual codes are contour mapping, color coding of electrical systems, and auditory tone distinctions used in telephone systems. Venn diagrams are used to represent inclusion—a notion that is difficult to communicate through natural language, especially when more than two sets are involved. Flowcharts provide visual representations of connectedness and sequence. In musical notation, symbolic codes (the forms of notes themselves) give specific information about pitch (through the position of a note on the staff) and duration, and the perceptual effect of “blackness” on the staff provides general information about the speed of the music.

Such principles of information display as the effective use of perceptual and symbolic codes can be applied to a variety of problems in interface design. Successful graphic and layout techniques can even make computer programs easier to visualize. For example, Fitter and Green [1979] have found that understanding is enhanced when different programs are represented in ways that are as perceptually distinct as possible. Baecker and Marcus [1989] similarly have applied knowledge about information display to the task of visualizing programs written in C.

Contemporary Interface Design Tools: Metaphor and User Observation

In addition to the wealth of wisdom to be gained from the study of design in other traditional domains, the discipline of interface design has developed some methodologies of its own that are worth reviewing. Related to the notion of borrowing techniques from other domains is the idea of creating interface metaphors that can anchor users' understanding of the computer to something with which they are already familiar. Conversely, studying people as they perform familiar, well-understood tasks can provide valuable insights about possible ways to represent tasks on the computer.

INTERFACE METAPHORS

Interface design is a relatively new human endeavor and has benefitted much from the application of metaphor in helping interface designers who understand it mainly by virtue of being human:

The metaphor is perhaps one of man's most fruitful potentialities. Its efficacy verges on magic, and it seems a tool for creation which God forgot inside one of His creatures when He made him. [y Gasset, 1925]

Metaphors are powerful verbal and semantic tools for conveying both superficial and deep similarities between familiar and novel situations. We are generally unaware of the extent to which metaphor is embedded in our thought and language (see Erickson's chapter on metaphor in this volume). As MacCormac describes them, "Metaphors not only communicate suggestive and expressive meanings but they also become iconic objects through their fusion of sense with sound" [MacCormac, 1985].

The Macintosh computer interface is a well-known example of the successful use of various metaphors to facilitate new users' understanding of the computer and its functions. During the development of the Lisa interface, Apple designers found that the "desktop metaphor" was preferred by users to a form-fill-in dialog box. Much of the ease of use of the Macintosh interface is attributable to the correspondence between the appearance, uses, and behaviors of such interface objects as documents and folders and their real-world counterparts; conversely, user difficulties are often attributable to differences between them.

Although some find the use of interface metaphors to be too constraining (see Nelson's chapter in this volume), designers can optimize the cognitive value of metaphors and minimize difficulties through careful selection and implementation. As Lakoff and Johnson [1980] observe, metaphors do not imply a complete mapping of every concrete detail of one object or situation onto another; rather they emphasize certain features and suppress others. Thorough examination of a candidate metaphor helps designers to discover its inherent selectivity and use it to their advantage.

OBSERVING REAL-WORLD TASKS AND ENVIRONMENTS

Development of some worthy "new" ideas for introduction into the electronic world can come from careful examination of the "traditional" ways of performing tasks. As Don Norman exhorts elsewhere in this volume, designers must sit down in offices, factories, and schools and watch the problems people encounter and the tools they use to solve them in their everyday work and education.

Observation of a person at a traditional physical desk reveals typical and extended ways of working that are not supported by a computer system. For example, the person may start writing a small Post-it™ note to carry with them, then on another plain piece of paper draw a map sketch, and on another plain piece of paper paint in different colors. Today, trying to do the same thing using a computer is immensely more complicated. Typically, different applications have to be launched separately depending on the kinds of data to be created; writing, drawing, and painting are all separate applications and different data types. Individual elements must be pasted together, sometimes requiring that they be transformed into new data types, in order to create a finished document.

The observations above have led to the further refinement of a set of interface ideas often referred to as the *plain-paper metaphor*, likening the screen to a blank sheet of paper. Currently, users must break their work up into separate tasks and perform each within its associated environment—taking the tasks to the tools. The plain-paper metaphor would reverse this process by bringing the tools to the task. "Virtual" tools could be designed to correspond to "real-world" uses.

Metaphors have two distinct but related uses in interface design: as cognitive aids to users, and as aids to creativity for designers. Metaphors can help designers to use their own, often unconscious expectations to create new information links and mental structures. The creative use of metaphor and other techniques for formally facilitating creativity are reviewed in the next section.

Techniques for Generating New Ideas

No one really knows where good ideas come from or what actually makes them "good". Novelty is desirable, and so is a close fit between the idea

and some real-world problem. Leverage and generality are also important. So where do these novel, useful, high-leverage, general ideas come from? What follows are some suggestions on how to stimulate thought by looking at the world in unusual and nontraditional ways.

Some people believe that new ideas are almost always the result of collisions—juxtaposition or recombinations of ideas [Koestler, 1964]. Consciously creating juxtapositions of normally unassociated referents may suggest new, previously unsuspected hypotheses about how to approach or solve a problem. Adams [1986] provides a series of brainstorming steps that can help designers expand existing metaphorical elements and generate new ideas. Designers can be encouraged more formally to think more broadly and openly about their interface designs by engaging in lateral thinking exercises. What follows is a series of interface design exercises based on Adams' axioms, many of which illustrate ideas that have been considered in the design of past interactive systems.

New Uses for the Object Consider the “desktop” interface. What else could a desktop be used for? People often use a real desk as a space for communication—laying out papers and showing intermediate stages of their work, marking them up together, putting them in a different order. The “desktop” could be used similarly as a space for shared work and as a communications area.

Adapt the Object To Be Like Something Else What if we adapt our “desktop” to be more like a kitchen? In kitchens, we find lots of basic and specialized tools that are used to create a myriad of different items (food). The same ingredient creates different foods when different tools are applied to it. For example, grating a carrot can produce fine-grained garnishes, but placing it in a blender produces juice. If we could identify or create some similar basic ingredients to serve as building blocks, they could serve as models for building better general-purpose tools. How would these ideas change the “contents” and organizational structure of an interface?

Modify the Object for a New Purpose What happens if we add the desire of providing information about the larger environment to our “desktop”? We might employ environmental sound, as in a real office, to provide information about adjacent activities. We might treat windows as periscopes or portholes through which we can observe or even travel to other parts of the environment (rather like Smalltalk project Views/Rooms from Xerox PARC).

Magnify—Add to the Object We could add various features to the “desktop” that are already present in a real desk; for example, a drawer with scissors, glue, tape, and other tools (PARC Cypress, 1977, and Apple II

Jane, 1983, are examples). We could add features like phones, calendars, diaries, FAX, newspaper service, or shopping lists. These items would enhance and extend its functionality.

Minimize—Subtract from the Object We might pare away the desk until all that is left is the “desktop” and a drawer with blank sheets of paper and a pencil in it. What could be accomplished with only these items? (This is usually described as the “plain paper metaphor,” an idea that may actually predate the notion of separate applications.) This helps prevent the mind-set of always adding more and more isolated pieces of technology.

Substitute Something Similar A “desktop” may not be a useful metaphor for every user in every task domain. What about a delivery service? Instead of things like files and folders, the interface could be organized in terms of trucks, routes, and ordering systems. The trash can could become a dumpster. Another interesting metaphor is the idea of an interface with peel-off transparent layers.

Rearrange the Data What if we reorganized the basic layout of the “desktop” so that the scroll bars were at the top (as in Smalltalk) and the menu bar was at the bottom? Among other things, we may discover such biases as a layout for right-handed people. We might also use the same technique to generate ideas for designing an alternative interface for left-handed users.

Reverse or Transpose the Information Turning the metaphor inside out creates some interesting ideas. What would happen if windows were views from files to the desktop rather than the other way around? What if everything were circular rather than rectangular in form? What if we were to view the desktop from below rather than above? What if we saw it from very *far* above—what else would come into view? Ideas of different visual lenses come to mind; for example, fisheyes and magnifiers.

Combine the Data into an Ensemble What larger metaphor might the “desktop” be a part of? Beyond its area is perhaps a virtual office, a whole building-full of data-rooms, cities, or worlds of information. How might this larger metaphor be represented? How would the desktop fit into it?

ROLE PLAYING

In addition to Adams’ exercises, thought provocation through role playing [von Oech, 1986] might help designers to create, evaluate, and develop ideas by assuming different points of view. Often it seems that the only point of view we have time for is the defense of our own existing positions, roles, and ideas. It is too easy to get blinded by our own myopic view.

Deliberately assuming different roles can be both creative and liberating. Von Oech describes four characters that are active in all of us to varying degrees and, depending on our job descriptions, that are allowed in different proportions to work on a problem:

The Explorer: This character gathers information on an issue, researching the problem before the solution set exists. Activities include reading, asking others about their views, and deciding which issues need additional work or definition.

The Artist: This character generates new ideas in the problem-solving phase. This phase is the most energetic and active. New problem definitions, potential solutions, and alternative next paths for action are produced.

The Judge: This character evaluates and filters the ideas that have been generated. At this stage some ideas must be discarded—a task that is less appealing to creative folks. New ideas won't emerge if the Judge is in charge at the beginning of the brainstorming process.

The Warrior: This character champions a particular idea and sets the course for the next round of problem-solving. This includes planning how the idea will be tested, evaluated, and developed.

Creative problem-solving can be enhanced by exercising mental agility—the ability to move easily between different roles. No one of these four types of characters can make an idea succeed without the other characters' participation. Knowing how and when to change character roles is crucial. A useful exercise is to take 30 minutes to brainstorm and to plan out a solution to any particular problem, alternating roles between the four described characters every 7 minutes. This approach can demonstrate how each of the different roles contributes to the overall decision. We have tried such exercises at Apple and have found that the transcripts of such role-playing sessions offer much food for thought even later.

ENHANCING CREATIVITY

Is creativity a talent, accessible only to a gifted few, or is it a skill that can be learned? This question has been debated and researched from a surprising variety of angles, from discussions by theorists such as Koestler [1964] and Harman and Rheingold [1984] to practical brainstorming and unblocking guidebooks from applied imagineers such as von Oech [1986], Adams [1986], and McKim [1972]. Current research into the cognitive underpinnings of human learning is making important contributions to the study of creativity. Books and exercises for enhancing creativity are worthwhile for interface designers, especially those connected with visual thinking [Hanks and Belliston, 1980] and right brain exercises [Edwards, 1986]. They all serve as platforms for individual and group exercises that can help us better understand our users' needs and propose some usable and creative solutions to their problems.

As software becomes a more powerful and pervasive tool for communicating and learning, the need will increase to find new ways to look at interfaces and their designs. Interface designers can learn from the past successes and failures of other design/communication disciplines. The idea of *visual thinking* as promoted by Arnheim [1969], McKim [1972], and others emphasizes the success designers can achieve by educating themselves in new perspectives. The development of new prototyping tools and hardware platforms with greater capabilities, as well as techniques for using traditional design knowledge with new creative methods and fresh perspectives, will facilitate the design of exciting new interfaces.

Interface designers must live in both the present and the future. While we are working on our present interface problems, we should be thinking about how to create tools to better build on our knowledge and make it reusable and available to others. The best contribution that we can make to the understanding of the communication interface is to create and build tools to be better able to do our own jobs. Right now we are just beginning to discover what these tools should be, creating specifications for their design, and experimenting with new methodologies. In the future, we will be challenged to make the most of this power to pass on our understanding both to other designers and to new users in the form of new, more powerful, more easily learnable, and more engaging ways to use computers.