

Art and the User Interface¹

sean dreilinger

December 12, 1993

¹Written for a course in Human Computer Interaction & User Interface Design at the UCLA Graduate School of Library & Information Science

Contents

1	Introduction	2
2	What role have artists played?	2
3	The first Icons...	3
4	What do artists think of aesthetics in existing interfaces?	4
5	What do HCI experts think of aesthetics in existing interfaces?	5
6	The Artist–Programmer Dichotomy	5
7	Contemporary Icons?	5
8	Future directions for Art and Interface...	6
8.1	Document and Standardize Visual Interface Elements Across Platforms...	7
8.2	New Perspectives On GUI	7
8.3	Design Process	8
8.4	Healthy Competition	8
8.5	Creating the Illusion of Consistency in GUI Software...	8
9	Interdisciplinary Conclusion	9

1 Introduction

Ubiquitous computing... little computers everywhere... the idea comes from Art. Society has been benefiting from *Ubiquitous Art* for years now. Something in almost every environment you enter today has been designed, improved or influenced by an artist. Look at your clothing—some of it must have been designed by a fashion artist. Look around the room—the chair you sit in, the desk you sit at, the telephone you talk on, the picture frame that surrounds a family photo, the utensils we eat and cook with, the jacket of the book we read, automobiles, postage stamps, the advertisement on the back page of a magazine, even the very typeface of the paper you’re reading now. Art enhances and improves our aesthetic environment at every opportunity—until we sit down at a personal computer.

Blah!¹ Something is not quite right with the aesthetics of today’s graphical user operating environments and software applications. As a formally trained visual artist, I want to learn what role Art and the artist play in the creation of GUI software in the past, present, and future. My three main questions are:

- How have visual artists contributed to existing user graphical interfaces?

¹Not always, of course, but we’re going for the dramatic beginning

- What do people think of aesthetics in existing GUI’s?
- How can visual artists and the discipline of visual art help improve graphical user interfaces now and in the future?

I’ve prepared for this discussion in the following way:

- By reading books and articles that address artistic aspects of user interface design. At every opportunity, investigating icons as a graphical element of focus.
- By speaking with practicing commercial artists who rely on computer software to perform their job.
- By completing two related assignments for a course in Human-Computer Interaction this quarter, and...
- By brainstorming some potential applications of Art to interface design for our ‘future contributions’ section.

2 What role have artists played?

As a student and worker whose primary research, work, and recreational tool is accessed through the glass pane of one 14” video monitor, I am sensitive to the graphical presentation of the software that I interact with all day (& all night) long. Some interfaces appeal to me because they *feel* consistent and are satisfying to use. Others impress me as utter crap. In a windowed Graphical User Interface (GUI), there might be several functionally equivalent text–editing programs on the screen at once—each displayed within a window. One window might contain an editor with a tacky, awkward user interface. Another editor, visible at the same time, might feature a gracefully designed, awesome user interface. Four different editors are arrayed below: Assuming their underlying capabilities are equal, I will

Figure 1: 5 Interfaces to the VI Editor

naturally be drawn to the interface that has greater aesthetic appeal. The GUI operating environment further influences the my experience by literally ‘framing’ each running application’s interface into a window, the functions and limitations of which will vary with each operating environment—windows in X–Windows have features that are different from windows on a computer running Macintosh System 7; windows in OS/2 are displayed and manipulated differently from those offered by Microsoft Windows NT.

How much have these aesthetic differences have been determined or influenced by visual artists? If my decision to use or purchase a software package might ultimately come down to ‘how good it looks,’ why are so many feature–rich graphical applications an artist’s view of chaos? According to Theodor Nelson, the ‘Art’ we’ve seen in user interfaces to date has been art created by programmers, for programmers. . . and it is not based on the same principles that seem intuitive and aesthetically pleasing to end users. . .

On Artistry: Historical accident has kept programmers in control of a field in which most of them have no aptitude: the artistic integration of the mechanisms they work with. It is nice that engineers and programmers and software executives have found a new form of creativity in which to find a sense of personal fulfillment. It is just unfortunate that they have to inflict the results on users.¹¹

When the Xerox PARC group published their work on the Star user interface, they described their attitude towards the *process* of user interface development as “an art.” It was good to read that from the very beginning, and in the most controlled research environments, certain issues have been ‘scientifically determined’ to be a toss–up, where the artistic judgment of the designers should take precedence over research data.

User interface design is still an art, not a science. Many times during the Star Design we were amazed at the depth and subtlety of user-interface issues, even such supposedly straightforward issues as consistency and simplicity. Often there is no one “right” answer. Much of the time there is no scientific evidence to support one alternative over another, just intuition.¹⁴

Viewing screen captures from the Star system is not as easy on the eye as working with some of the graphical interfaces we have today. Given the limitations of their display hardware, its pretty good looking.² A particularly interesting theory throughout the development of the ‘desktop’ metaphor has been the idea that the desktop serves as an extension of the user’s short term memory—a user might refer to the information on screen the same way they would roll their eyes back into their head and ‘glance’ for a bit of information in part of their brain.

A well-designed computer system can actually improve the quality of your thinking. In addition, visual communication is often more efficient than linear communication: “a picture is worth a thousand words.”¹⁴ The Xerox engineers’ observation that visual information is communicated quickly comes as no surprise to visual artists. I’ve read several anecdotes relating the first thing programmers did when they connected computers to video monitors—they merely recreated a teletype printer on screen, with all of its quirks and limitations! In promoting the use of a visible interface, the Star group repeated “a picture is worth a thousand words,” which was probably an appropriate use of the clichè. I would add to this that those ‘words’ will vary according to how well the interface is designed. A good visible interface will be worth a thousand words of poetry or well written prose. The thousand words worth expressed by a crummy interface might be more like graffiti on a bathroom wall or cursing someone out—definitely the kind of ‘noise’ an end user will take pains to avoid.

3 The first Icons...

They chose their pictures well... well, *scientifically*. The icons that were eventually included with the Star Office Workstation were carefully engineered to be recognized quickly by the greatest variety of users. . . In fact, “the [overall] design effort took more than six years... The actual implementation involved from 20 to eventually, 45 programmers. . .”¹ The ‘icon designers’ mention no aesthetic considerations whatsoever in their drawing, testing, or selecting of icons to be included in the Star System. Still, these were the best graphical user interface icons ever developed—

²For a bunch of engineers trying to draw... snicker :)

they were the only ones. Today, hundreds of additional icons have been tried in graphical applications to represent the same objects and functions. Many of the preferred icons in use today bear some resemblance to their ancestral Star bitmaps. The role of the

Figure 2: Original Xerox Star Icons

visual artist in modern user interface design is always increasing. Teams working on big GUI projects are *expected* to include an artist on the design team from the very beginning. Too many times in the past, the visual artist has not hired to help design a user interface until a conflict arises within a design team. When software is designed by a committee, and an internal disagreement arises over how some graphical element of an application should be implemented, graphical artists are "...often treated as 'firefighters'—called upon in an emergency to repair disasters not of their own making, under severe time constraints."¹⁹ Obviously, this is not giving end users the benefit of an interface that has been visually conceived from day one. Experts such as Bruce Tognazzini of Apple Computer highly recommend that at least one visual artist participate in every user interface design project from the outset:

Graphic designers should be brought in at the beginning of a project, not the end; until the rest of the team sees the designer in action, they will not think visually. Those of us who lack the talent to draw have long since learned to avoid coming up with ideas that require drawing! Once having been exposed, people will start thinking visually and will get the habit of doing quick place-holder graphics, to be worked up and worked over by the graphic designer later on.¹⁶

At the same time that there is an effort on the part of other team members to participate actively in the visual design, graphic artists should brush up on the cognitive psychology research that has been done on vision, icons and recognition, symbols, and semiotics.

4 What do artists think of aesthetics in existing interfaces?

Most artists *don't* think about user interfaces, unless they are involved in designing one. At least that's the impression I got from speaking with a team of commercial artists at Creative Marketing Directions, an advertising agency in San Diego, California.³ When I asked them what they would improve about the user interfaces they interact with all day long, their first and unanimous reply was 'bigger, non-glare monitors.' Many of the items at the top of their wish-list were hardware-related. Other top-priority items included ergonomic keyboards and increased color depth for some machines. These requests come from experienced artists who log long hours of image editing and perform graphic document layout on state-of-the-art³ Macintosh equipment. Amit Shamis, an intern in accounting, wanted the firm's one IBM-DOS system to display a menu of available software at boot-up.

When our conversation focused on the actual graphical user interfaces, three points came through as areas of concern with these artists: Andrea Frankeiwicz, a Creative Services Manager for the agency, wants better documentation, which goes beyond the cursory 'getting started' section to provide tips on advanced use of an GUI applications such as Quark Express or Photoshop, *and* covers how to make related programs work together. For example, creating one advertising brochure might *link* elements created in Freehand, Photoshop, Microsoft Word, and Quark. Additional software must be used to compress the resulting binary file and output it at a service bureau.

Paul Billimoria wants all of his related files from different applications *automatically* created in and saved to a 'project folder.' This would save him the time and hassle of locating these documents manually. This idea drew raves from his coworkers. Finally, we discussed error messages, which we all agreed are hopelessly cryptic in DOS and are still not informative enough on Macintosh. I got the feeling that these smart users could take some steps to prevent their computers from locking up or crashing, if only the interface would give them some kind of feedback about why it was stop-

³*state of the art* is the term that caused the most false drops during my early online research for this topic: *art* and user interface.

ping. Ann Lewin, also in Creative Services, suggested that the computers be outfitted with the equivalent of an automobile’s dashboard warning lights—something that can tell a user about a problem before it arrives by surprise

My visit to Creative Marketing Directions taught me that there is some minimum level of hardware and site quality that are more important to the user than the software interface. If one has to squint all afternoon because sun is streaming into the office and onto the monitor, that user is literally in no position to discuss the finer aesthetic impact of their software’s interface. Physical comfort and compatibility must be achieved before an interface can be evaluated. I expected a group of working artists to have a number of ‘aesthetic peevies’ about their mission–critical user interfaces... I was wrong. Instead I heard tales of frustration almost identical to those I hear among users in the GSLIS Lab. Maybe interface aesthetics are not on anybody’s mind in such a conscious way.

5 What do HCI experts think of aesthetics in existing interfaces?

If the artists aren’t thinking about it yet,⁴ Who is? In the literature of graphical user interface design, Opinions on good–looking GUI’s ranged from enthusiasm to resigned acceptance. Bruce Tognazzini, living up to his job title of ‘User Interface Evangelist,’ tackles the aesthetic design of interfaces with the attitude that “We, as designers, are not simply assembling dialoguing systems, we are building worlds, carefully crafted virtual realities, in which users will live and work.”¹⁶ A user would probably enjoy working all day ‘in’ an interface designed with such a sense of responsibility and enthusiasm—especially compared to working in an environment constructed from the following point–of–view, in which the author sounds like he is stubbornly giving in to the concept of an aesthetically pleasing interface: “Now the technologies of computer display systems have improved, and the user population has expanded to such an extent that it is now advisable to include ‘attractiveness’ as a tool for improving usage of information systems.”¹⁸ Perhaps this point of view stems from the ancient feud between programmers and artists.

6 The Artist–Programmer Dichotomy

The explanation was simple: computer scientists and graphic designers have different priorities. Computer scientists value the program and how it works; graphic designers value the picture and how it looks.⁸

Individuals involved in interface design recognize a dichotomy between programmers and graphic designers. As members of a design team, the two disciplines often have seemingly opposite priorities—the programmer wants to get the darned thing to compile, and doesn’t give a hoot about how it looks yet. The programmer is willing to stay extra hours until the code is working well. The graphic artist will stay up late trying to achieve a harmonious layout of graphic elements, but it doesn’t matter which features are and aren’t compiling—none of it will be any good if the user can’t interact with it on the screen (or tablet, or kiosk, or wherever).⁸ What the two *team* members often forget is that they are united in the common cause of delivering an efficient, speed–optimized, easy to learn–and–use software package to a group of users. Those end users probably don’t care whether or not the program works because the code is so well written or because the graphical elements are so well thought out—as long as it works well.

The artist and the programmer are the *yin* and *yang* of interface creation. They constantly spin against one another, but the result is a happy balance between form and function. Artists of GUI (if there are any, yet) *need* the work of genius programmers to support and give meaning to their art. Genius programmers *need* communicative graphic artists who can *interpret* technically brilliant ideas and *translate* them into recognizable icons and other GUI elements so that a non–technical user base can appreciate (and purchase) them. Programmers and interface artists have a symbiotic relationship: they *need* one another.

7 Contemporary Icons?

To get an idea of what aesthetic considerations go into the designing of a present-day icon, I consulted the Software Development Kit (SDK) documentation for a number of operating environments. Surprisingly, two of the newest SDK’s offered little or no advice for

creating icons. These were the *Microsoft Win32 Software Development Kit Final Release, Version 3.1*, and the *IBM OS/2 2.1 Professional Developer's Kit* from March, 1993. The two kits, for writing Windows NT and IBM OS/2 programs, offered technical chapters on coding icons, but no discussion of how or why they should be used. The style Guide for the *Open Look GUI* environment for UNIX advises that final versions of all icons should be drawn by a professional artist. The *Open Look Style Guidelines* ask developers to perform a sort of 'ico-analysis.' When the icons created for an application in the Open Look GUI meet (or don't meet) certain criteria, the developer should consider options such as combining or separating programs, designing an entire 'family' of visually related icons, or incorporating animation into their icon design.¹⁵ In their own version of a SDK for Microsoft Windows, Borland has the good sense to devote about one-half of a page to icon design issues. They ask the software designer/programmer to answer three questions before designing an icon for any Microsoft Windows application:

- What should the icon represent to the user?
- Is your design simple enough to fit into a 32x32 or 32x16 pixel area?
- Will the icon look good no matter what background the user chooses to display it on?⁶

From the SDK documentation that I could get a hold of for this paper,⁴ very little guidance for aesthetic design was offered. Borland devoted a half-page to the aesthetic considerations of icon creation. The printed documentation for the Borland SDK weighs about fifteen pounds! "Icons in Everyday Life," a paper by William and Susan Wood, notes that even when specific graphic role-models are provided, today's developers disregard the carefully considered guidelines in favor of their own. The Macintosh interface guidelines offer artists some simple 'meta-icons,' on which to base their own designs. Although some designers have adhered to the framework, many more have strayed. Here is the table presented by the Woods:²⁰ Other aesthetic considerations for each operating environment were also omitted from their respective documentation—and this

⁴NeXTStep (drool!), Macintosh System 7, others weren't available to me.

Figure 3: Macintosh Icon Style Guide

made something clear. When designing the visual interface for a software application in an operating system or environment, the designer is *restricted* by the standard graphic elements of that system. While custom code could probably be written to support almost any visual feature, it is much quicker to simply 'settle for' the graphic elements or 'widgets' already provided by the SDK author.

The lack of aesthetic guidance for software development was evident during a fun exercise in which our HCI class visited a Macintosh computer lab and played with shareware software. In the course of an hour, we sampled dozens of home made and semi-professional software creations. While those 'constant' graphical elements of the operating system were usually visible in each design—scroll bars, menu bars, title bars, etc.; other interface elements were all over the map. Some icons were way out of proportion with the others—as though they had been stolen from their X-Windows siblings. Other applications had elements on the screen that were drawn in shaded 3-D perspective with the imagined light source coming from the upper left, while others put the imagined light source elsewhere (like from the bottom right!). To really see things pop on your eyes, try running two applications like that side by side and staring at them for a few minutes. Other amusing aesthetic designs included self-defeating foreground text-background color combinations that were unreadable, and 'illustrated' adventure games with animated illustrations that ranged from 'interesting' to downright ugly. Visual artists everywhere should perform some community service by hooking up with a local shareware author and helping them rethink their GUI's.

8 Future directions for Art and Interface...

Visual artists and the field of Visual Art have a lot to offer to the design of graphical user interfaces. Here are five contributions that visual artists can make now and in the near future...

8.1 Document and Standardize Visual Interface Elements Across Platforms...

At present, various software development guidelines offer little or no instruction for creating an artistically integrated user environment. Instead, they rely on the built-in elements of the interface and advise programmers: ‘hire a professional artist to design your graphical features.’ Research on the way people process visual information is conducted in multiple fields, including Art, Medicine, and Psychology. Someone needs to compile this knowledge into guidelines that can be used by *working* artists and programmers designing GUI software for every platform. Imagine the trouble developers must face today when they attempt to port a program from one operating system to another. Because each operating system is based on entirely different graphical standards, there is little common ground and designers are forced to either compromise their original program or spend long hours recreating the features of one operating system in another. Consider the benefits for users and programmers alike if we could have standardized icons across GUI’s...

Figure 4: Variations on the international ‘I’ for Information Icon

Standardized Icons Many computer operating systems perform similar functions on similar objects—a user of command-line Vax, UNIX, DOS; or graphical X–Windows, Open–VMS, MS–Windows, and Macintosh will still ‘copy’ files, ‘delete’ files, create ‘directories’ or ‘folders,’ and connect to other machines on a network using invisible protocols. Why not settle on some well-designed, *standardized icons* to represent these objects and actions in all graphical environments?

A major force that opposes any effort to standardize many computer icons is the desire of many developers to be unique. Every designer wants his application to stand out, and quite frankly, it is more fun to design a special set of icons without trying to imitate others. Often a company wants to reflect its

corporate logo in its icons. There has also been an effort by some creative designers to copyright striking and innovative glyphs. It is the user who suffers most from the lack of conformity on individual computers and between computer systems.²⁰

Standardization works in airports, train stations, and on traffic signs all over the United States and Europe. Maybe designers don’t understand the benefit that such standardization could have for users—especially those who must work with more than one application or operating system. If the manufacturers won’t agree, perhaps the artists can.

8.2 New Perspectives On GUI

One role of the artist is to show the audience something familiar in a new light... Marcel Duchamp did this by putting things like snow shovels, bottle racks, and urinals on pedestals in art galleries. John Cage did this by composing pieces of music which required an audience listen to the sounds of silence (and itself) in an auditorium for four minutes. While few consumers today hang urinals on their walls or buy CD’s with no music on them, the Art that they do purchase en–masse is *influenced* by the work of cutting edge artists—who constantly reinvent and reinterpret their world. As more and more of us work in a world where people communicate through computer interfaces, we will need someone to help recreate that work–environment from time to time, lest it become so predictable that our quality of work (*productivity*, in buzz–terms) begins to wane. The visual artist will be an ideal contributor to this ongoing re–creation and re–thinking of existing visual interfaces.

For an idea of what I mean about re–thinking the interface, consider Brenda Laurel’s book entitled *Computers as Theatre*, which starts by discarding most of our assumptions about human–computer interaction and replacing them with concepts and ideas borrowed from the theatre.¹⁰ Will HCI professionals buy this? I haven’t seen anyone adopt it outright, but with the increasing development in *multimedia* and *virtual reality*, Laurel’s viewpoint presented here could be a big *influence* on the developers. They might even adopt a new...

8.3 Design Process

From taking an HCI course, reading articles, books, and SDK documentation on user interface design, I see that ‘good’ user interface designs begin with research into users, their tasks, and the features to be included in the software; they end with an attempt to integrate scientifically-chosen user interface elements within an artfully crafted, commercially appealing package. An artist creates in the opposite direction—from the inside out. An artist creates a work, based on something inside—an intuition, a feeling, belief, purpose, an inside sense about the world around them. *After* the artist has ‘finished’ a work, viewers and critics will evaluate it. Depending on the type of Art it is, it may or may not be revised and improved. A motion picture crew can shoot a scene, view and evaluate it, and redo the whole thing if needed. Other works of art can not be repeated—as Joni Mitchell the folk singer tells members of her audience who request she play certain songs: “I wonder If anyone ever asked Van Gogh—‘Hey, paint *Starry Night* again, man!’” By analyzing ‘successful’ works of art, patterns or elements that predict repeatable success can be derived and applied to future work, without violating the artistic integrity of existing work.

The integration of software can not be achieved by committee, where everyone has to put in their own addition... It must be controlled by dictatorial artists with full say on the final cut.¹¹

The connection between the artist’s inside-out design process and the committee’s outside-in process could be made by an individual assigned the task of ‘aesthetic integration’ for a project. It need not be a ‘dictatorial artist,’ but ideally a person who came up with the original inspiration for the project or someone who shares wholeheartedly in this motivation. Inside-out design may be used to augment existing software design practices. Contests are great tools to get artists working on something. Imagine the fun and ideas a software design team could generate by sponsoring a design contest for the visible elements of their interface.

8.4 Healthy Competition

The Museum of Modern Art should offer awards for excellence of user interface. Donald Norman, author of

The Design of Everyday Things, says he dreads the day these awards are handed out,¹² but I think it would be a step in the right direction. It would validate interface aesthetics to designers, give them an added incentive to consult with an artist before creating an application for a visible interface. Ben Shneidermann encourages his students to design and try out competitions among user interfaces.¹³ If authors and manufacturers know that the aesthetic value of their product will be evaluated by a panel of experts and reported to potential users, they will be forced to make improvements. We will also see that improvement when today’s computer-oriented publications that write reviews and comparisons of hardware and software add an ‘artistry’ or ‘aesthetic integration’ factor to their testing and evaluation articles.

8.5 Creating the Illusion of Consistency in GUI Software...

Jonathan Grudin has observed that sometimes the most ‘intuitive’ or *seemingly* consistent behavior for an interface actually *violates* the pattern of continuity established by the program’s designers. “... the ‘inconsistent’ interface choices described earlier are so naturally molded to the users’ tasks that users may not even notice the inconsistency.”⁵ Design teams need someone to be an advocate for the user, someone who will continually discourage programmers from representing program features to the user in technical terms. A visual artist is the ideal candidate for this position.

Artistic manipulation can be made in an interface to create the illusion of consistency—Art on paper can create the illusion of depth when there is none. Art in animation creates the illusion of movement when there is none. Visual Art in general creates the illusion of visual harmony (or discord) within a narrow time frame—and does it well. If user interface designers can persuade artists to apply these elusive talents to the user interface, programmers would be able to incorporate unique and powerful features into their application, and users would *still* gain the speed and ease of learning associated with consistent interfaces—because a visual artist would be involved in the project to help assure a consistent look and feel.

9 Interdisciplinary Conclusion

The way I've been hyping visual artists throughout this paper, you might think that all you need to create a top-notch graphical user interface is a top notch visual artist. To tell you the truth... I don't think it works that way. Visual Art has been an obvious element for consideration in user interface design since the introduction of the first GUI, yet artists are still viewed as an 'outside consultant' to a team of (more important) programmers. What we really need is better understanding, acceptance, and cooperation among the various disciplines that contribute to GUI design, *including* Visual Art. Scott Kim envisions a sort of 'Interdisciplinary Ambassador'⁸ who can bring together experts from diverse disciplines and help them pool their resources to create results greater than any other combination or individual could achieve:

Together, a graphic designer and a programmer can create something much stronger than either one could create alone.⁸

Visual Art has been a field of study for hundreds of years. Human-Computer Interaction is a baby by comparison. Why reinvent the wheel? Developers of today's and tomorrow's Graphical User Interfaces should stand atop the mountain of knowledge and experience offered by the field of Visual Art, and continue to climb from there.

References

- [1] William L. Bewley. Human factors testing in the design of xerox's 8010 'star' office workstation. In Ronald M. Baecker and William A.S. Buxton, editors, *Readings in Human Computer Interaction: A Multidisciplinary Approach*. Morgan Kaufmann Publishers Inc., 1987.
- [2] H.M. Deitel and M.S. Kogan. *The Design of OS/2*. Addison-Wesley Publishing Company, Menlo Park, 1992.
- [3] Sean Dreilinger. Creative marketing directions interview, 1993. Discussion with the staff of an advertising agency.
- [4] John Gantz. A matter of timing: Fine arts share of the computer graphics market. *Computer Graphics World*, 13(7), 1990.
- [5] Johnathan Grudin. The case against user interface consistency. *Communications of the ACM*, 32(10), 0000.
- [6] Borland International. *Resource Workshop User's Guide*. Borland International, 1987.
- [7] Alan Kay. Dr. alan kay: Doing with images makes symbols—communicating with computers. video cassette produced by Judith Lemon and Karen Mathews, 1987. SFO Productions, 100 minutes.
- [8] Scott Kim. *The Art of User Interface Design*, chapter Interdisciplinary Cooperation. Addison-Wesley Publishing Co. Inc., Menlo Park, 1990.
- [9] Brenda Laurel, editor. *Art of Human Computer Interface Design*. Addison Wesley Publishing Company Inc., Menlo Park, 1990.
- [10] Brenda Laurel. *Computers as Theatre*. Addison Wesley Publishing Company Inc., Menlo Park, 1992.
- [11] Theodor Holm Nelson. *The Art of User Interface Design*, chapter The Right Way to Think about Software Design. Addison-Wesley Publishing Co. Inc., Menlo Park, 1990.
- [12] Howard Rheingold. *The Art of User Interface Design*, chapter An Interview With Don Norman. Addison-Wesley Publishing Co. Inc., Menlo Park, 1990.
- [13] Ben Schneidermann. *The Art of User Interface Design*, chapter User Interface Races: Sporting Competition for Power Users. Addison-Wesley Publishing Co. Inc., Menlo Park, 1990.
- [14] David Canfield Smith, Charles Irby, Ralph Kimball, Bill Verplank, and Eric Harslem. Designing the star user interface. In Ronald M. Baecker and William A.S. Buxton, editors, *Readings in Human Computer Interaction: A Multidisciplinary Approach*. Morgan Kaufmann Publishers Inc., 1987.
- [15] Inc. Sun Microsystems. *Open Look Graphical User Interface Application Style Guidelines*. Addison-Wesley Publishing Co. Inc., Reading, Mass., 1990.
- [16] Bruce Tognazzini. *Tog on Interface*. Addison-Wesley Publishing Company Inc., Menlo Park, 1992.
- [17] Craig Upton. Visual programming in data flow environments. video cassette, 1992. produced by Silicon Graphics, Inc. and University Video Communications. 60 minutes.
- [18] Richard H. Veith. *Visual Information Systems: The Power of Graphics and Video*. G.K. Hall & Co., Boston, 1988.
- [19] Laurie Vertelney, Michael Arent, and Henry Lieberman. *The Art of User Interface Design*, chapter Two Disciplines in Search of an Interface. Addison-Wesley Publishing Co. Inc., Menlo Park, 1990.
- [20] William T. Wood and Susan K. Wood. Icons in everyday life. In Ronald M. Baecker and William A.S. Buxton, editors, *Readings in Human Computer Interaction: A Multidisciplinary Approach*. Morgan Kaufmann Publishers Inc., 1987.

i'd rather get a bad grade for an honest, late paper than an honest grade for an early, bad paper—sean dreilinger