re:skin

edited by Mary Flanagan and Austin Booth

ARTECA
From the MIT Press

MIT Press books may be purchased at special quantity discounts for business or sales promotional use. For information, please email special_sales@mitpress.mit.edu or write to Special Sales Department, The MIT Press, 55 Hayward Street, Cambridge, MA 02142.

This book was set in Janson and Rotis by Graphic Composition, Inc. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Re : skin / edited by Mary Flanagan and Austin Booth.
p. cm.
ISBN-13: 978-0-262-06260-2 (alk. paper)
ISBN:10: 0-262-06260-7 (alk. paper)
1. Skin—Fiction. 2. American fiction—21st century. 3. Canadian fiction—21st century. 4. Skin—Psychological aspects. I. Flanagan, Mary, 1969– II. Booth, Austin.
PS648.S5R4 2007
813'.609353—dc22

2006056675

10 9 8 7 6 5 4 3 2 1

15
RESKINNING THE EVERYDAY

Mary Flanagan

borders

Pushing at that percussive gel-filled screen in front of me, the screen's skin ripples slightly to my touch. This thin titanium box is a tree whose roots extend over and within the planet and into outer space. Webcams mounted on the Mars rover, SETI space blips link me to places I will never see with my own eyes.

I push at my skin, and am a little thrilled and disturbed to note that it too ripples slightly to my touch, having a similar timbre and viscosity as the screen. The largest organ of the body, the great container, the esteemed purifier, the separator between outside and inside, between smooth and ripply, between liquid and the concoction of gasses that make up the air around me. Another screen (figure 15.1).

These surfaces, these interfaces separating two phases of matter, form a common boundary and help me know where I am in the world. The surfaces of everyday life amid computational systems, however, make things less and less clear.

When I think about computing, I think about boundaries and boundary crossing, translation: transforming the world around us into data, and from data back into the world. This process seems more and more transparent, but the transitional practice continues, and the interface remains. Like the interior transition between thinking and speaking, thinking and typing or thinking and clicking are shifts as well, and these everyday actions bring about a range of questions about the borders between bodies and machines.

```
/*Sebastian asks me, Are you writing fiction or is this an academic
essay? Are interfaces directions to something, or something in and of
themselves?
*/
```

However we see the progression of these physical ways of interacting with the computer, interfaces play an instrumental role in shaping the conceptual experience of

| **Figure 15.1** |
The everyday Macintosh interface

computational worlds, and, indeed, the real world around us. This essay is a meditation on the everyday computational interfaces which surround us, offering four conceptual sites for consideration. Each site marks a symbolic progression in thinking about the way computers are integrated into our everyday lives. This document enacts a conversation, an experiment that offers as many questions as answers about the various ways we think about interfaces.

The links between people and computers are only understandable through our interfaces: both the site and the process of individuation among bodies, objects, tools, and concepts. The first interfaces to explore, then, are the physical, material interfaces of digital devices.

boxes

Is the screen speaking back, redefining itself, redefining the box? It seems as though non-Western ideas about interfaces and boxes could bring unique systems, where hardware and software could *reflect alternate notions of time, space, place. While our computers remain in boxes for the time being, they become ever smaller and faster, and it is more difficult to distinguish thinking from unthinking objects.*

So many computers in so many homes. In the suburbs my friends keep home offices in spare bedrooms and basement annexes. In the small apartments of the city, it is often the dining room or kitchen space that is refashioned for the computer's ever-presence. The tables hold more plastic boxes and power cables than lasagna and sushi. Eating over his laptop, Sebastian looks over to me to ask if I see any napkins. At night, friends gather around laptop screens, lovers bring them to bed to watch movies, the box piled on pillows in an intimate agreement. How many of us will have our computers tell us bedtime stories?

There was no one "inventor" of the computer interface. In the history of computing, dials and switches eventually gave way to keyboards based on typewriters and teletype machines. These were used to record data on punchcards for machines such as the behemoth ENIAC computer of 1946. Vannevar Bush, once director of the Office of Scientific Research and Development in the United States, is counted among early interface visionaries. In his 1945 article "As We May Think," Bush offered models for a variety of methods of input and output including desk-based computer systems, speech interfaces, and even a walnut-sized camera worn on one's forehead. Interesting and still experimental-seeming devices such as the light gun interface were later developed for use in air defense control systems. A profusion of working interface technologies was developed during the twentieth century. The light gun was followed by the light pen in the early 1960s. Xerox PARC researchers led in both software and hardware interfaces, and went on to originate many of our graphical user interfaces, including the mouse (figure 15.2).

While designers and engineers work to erase mechanical portals between users and machines, and science fiction writers fantasize about direct mental connections to computing systems, it seems the opportune moment to shift ourselves away from our perception of current, everyday computing experiences. We need to examine the interfaces to our virtual selves, these layers of skins, through the beige box and trailing puck—these still offer, for the most part, the common interface over the last twenty-five mass-produced years. If we think in Mandarin, the box, keyboard, and the mouse are our *guo yu*, our common, everyday language. While the box becomes smaller and more stylish, the box is still a box.

Donald Norman (2002), in his writings on user-centric interface design, insists that makers of objects and of virtual worlds create a conceptual model that users can understand—models that become a part of the user's intuitive knowledge as much as gravity or the properties of water. Designers work to provide "intuitive exchanges" with

| **Figure 15.2** |
Traditional interface

systems—any type of system that requires interaction. Examples include user-friendly escalators, doorways in buildings, shifting controls inside an automobile, or computer application interfaces. The argument for intuitive systems is culturally based, of course; for what constitutes an intuitive signal to one person in Taiwan—tapping the table to communicate that a tea cup is full, or knowing the shoe cabinet should be near or outside the front door—is distinctly different than an understanding of the same social practices in another culture. Interfaces are the means through which we take clues and signals in a given culture. Learning new interface systems changes our behavior and is one thing that makes travel so invigorating.

As a social practice, interfaces should be highly contested and constantly reforming sites of social negotiation. Yet the computer box paradigm is pervasive and its design colonialist in nature. Asian manufacturers and suppliers follow design trends that have been, for the most part, established by the West. Internationally, keyboards follow the English-language model of input and define how language is translated from hand to machine. Computer literacy around the world has meant that citizens adopt retrofitted beige or black boxes into their homes and community centers and pursue a Western style of work and leisure. Somehow, a mouse in one hand has become a naturalized method of geographical and conceptual navigation.

I encounter challenges in articulating the boundaries among computer worlds, systems, and the physical world all of the time, as such systems become seamlessly integrated into the surrounding physical environment. However, at institutions and among those environments less endowed with the latest technologies, "boxes" remain. The most recent reminder of the pervasiveness of the box occurred during an after-school session with middle-school girls; my team was designing a complex online system that appeals to underrepresented groups (figure 15.3). Visiting the Computer Clubhouse in Brooklyn, New York, for a design session, I asked a group of 11- to 13-year-old girls to draw a computer game world that I had just described: a world in which characters would want to care for others and do things together socially. The girls pulled out pens, pencils, and paper. Every one of the eleven girls sat down and drew the box, the keyboard, and mouse instead of sketching the fantasy world I described. The dominance of the box has infused cultural imagination to a point where it cannot be separated conceptually from stories and ideas.

The worlds we drift in and out of so seamlessly are interfaced through boxes and wires and graphic templates that categorize our work and play. I, for one, have to constantly remind myself of this. Perhaps it is some kind of dreamworld I enter when I wake up in the morning, but I no longer notice the device itself, its color and shape, how my hands reach for the mouse. Like signing, I speak with my hands. I do not remember that I communicate through typing or that I use a mouse or touchpad; in the same way, it is difficult to remind ourselves of framing and the limitations of our native language.

In the daily quest to transfer ideas from head to document, I often take these interfaces for granted. Already ensconced in the realm of possibility the computer represents, I function inside its unique conceptual framework. In other words, the technology has become as invisible to me as anyone else working with the machines, as invisible as pen and paper may have been fifty years ago. But I remind myself to constantly examine the kinds of ink we use, the shape of the pen, the economic systems that produce these pens, how the pen shapes thinking, and the reasons we use pens in the first place over other recording devices. In the real world, pens and pencils don't necessarily represent cultural systems and values, but I believe that computers do. Computers run by relying upon zeros and ones, ons and offs, hard drives gridded out in block parcels, software constructed in distinct hierarchies. Computers contain nested structures within structures, each drawing from a different discipline: engineering to design, architecture to literature. Once we are faced with a paradigm, however, the underlying assumptions

| **Figure 15.3** |
Local New York design partners

on which it is built become invisible. However, as philosopher Thomas Kuhn ([1962] 1996) suggests, particular paradigms can also be very useful, gaining their status primarily because they are more successful approaches than competing systems of organization; they allow inquiry and work to advance faster than beginning the learning process all over again. We pick up the appropriate conventions depending on the task at hand. Paradigms can speed up research, scholarship, and other forms of work, but at the expense of taking the underlying system for granted. Though computing has been in mainstream imagination for a relatively short time—the Internet for just a decade—the priorities and use of the system seem to go by unquestioned. Thus there is a tension between our current desire to challenge dominant systems and our need to make use of the efficiencies they provide.

Invisible interfaces may make computing, coffeemaking, and navigating tollways easier, but such invisibility may also mean that much of the U.S. public will be unable to participate in authoring culture. Computer literacy courses at many high schools across the United States are fundamentally typing classes in graphically organized software packages. Computing as an ideological system is not yet a topic of the humanities, and students are not taught to question and delve into conceptual systems the computer represents. It is no coincidence that the mystical and renegade role of the computer hacker surfaced in the 1980s as automobiles, phone systems, banks, and national security reached cohesive automation—the hacker mythos came at a time when the gaps that could undermine power relations seemed to vanish, where everyday tinkerers lost access to the mechanical aspects of daily life, and when the hacker, someone able to control a system—work in and around it—became the outlaw or artist crusader, a mythological hero/devil figure in mainstream imagination.

```
/*Sebastian asks me, What exactly is inaccessible about a computer?
They just follow human commands.
*/
```

Perhaps it is not so much the material that is inaccessible but the culture of computing itself, the guild-like feeling with its other language and presumed skill set from which everyday computer users feel alienated. While my mother, for example, could theoretically rework her operating system hierarchy to elevate the game of solitaire as the metaphor for her operating system, the perceived elite status and the specialized knowledge of those able to program computers affects an individual's perceived agency with their computers, especially in underrepresented groups. According to the National Science Foundation, the number of women earning bachelor's degrees in computer science in 1984 was 37 percent of the total, but by 1996 it had fallen to 28 percent of the total number of graduates. Women, as one category among many not represented in computer science, are not increasingly attracted to a discipline that guides much of the fabric of everyday life in technologically influenced (determined?) nations.

Currently there is growing research addressing such questions at the socio-cognitive and cultural borders of technological innovation. Implicit Association Task tests and Functional Magnetic Resonance Imaging studies continue to demonstrate that people have inherent biases down to physiological response in their recognition of images and words (Greenwald and Banaji 1975). Categorical tests on impressions of race and age consistently show that whites have a stronger same-race identification than do persons of minority groups in the United States. Other studies show that for a significant percentage of the population, men are associated with science, women with humanities; African Americans with violent and negative language; Asians with ability in mathematics (Phelps 2001; Phelps et al. 2000). Stereotypes, produced by culture, have been shown to become encoded into bodily responses to images. This means that at the very least, stereotypes are inadvertently encoded into our computational boxes and our everyday experiences, in icons, tools, and data structures.

Now that over half of all U.S. households contain computers with Internet connections, the daily lives of many citizen-consumers are bound to the computer. Every day computer users sit down, check their email, search for a movie trailer, read the news, balance a budget, or download digital camera images. The interface to the operating system of a computer sets the stage for the understanding and prioritization of data. After all, it is merely a representational tactic that "My Documents" are somehow different than *Adobe Photoshop*'s "Read Me" files. While data for the two sets of information may be stored in neighboring blocks on the physical hard drive, within the operating system's interface paradigm they are stowed worlds apart. We continue to believe our documents are special, our words and codes are somehow fundamentally different from all the other material filling the blocks of the hard drive. These constructed conceptual models are not neutral—interfaces (representing hierarchies and data structures) and the boxes carry meaning. For one, operating system structures emphasize an individual's separateness. Indeed, they emphasize individuality in general. Group-authored documents such as collaborative projects are not reflected in the structure of the system or in its iconography. Functions are separate from data; the network is conceptually demarcated as different from the local device.

How the box is shaped is of paramount importance. So is what it contains. The space of computational systems, and how these spaces are represented to users, is the second place to examine interfaces.

maps

The man climbed onto the fifth rung of the ladder, grabbed his hammer, and knocked a hole in my wall, plaster spinning everywhere in the air and across my shiny floors. He knocked a second hole next to the first, and dug his hand around the recess. A third hole was needed. One by one, the holes took over, my rooms effortlessly disappearing into the rubble of the moon . . . the walls a web of chalky river delta, a map held together with hair and dust. He promised that another would come by to fix it all up, to make sure that no evidence was ever seen, but of course I would always know. It was already past.

This is one of the lessons of Einstein's theory of relativity—looking inevitably means looking back, for light travels at a finite speed . . . If our center, our universe, is everywhere or nowhere, we do know that such a center could only be the present. We race from present to past at 186,284 miles per second. This text, that laughter, is perhaps a nanosecond away; the moon, almost two seconds lost. The present can only mean the lack of distance—a flash, the recognition of presence, found at the touch of an insect's wing on the top of the ear as it flitters off in a mid-

| **Figure 15.4** |
Mapping, this time: Stockholm

June breeze, or the mutual exhalation between interlocked bodies as the sun breaks to sear the dripping grass after a fresh-green shower.

Can I ever map such experiences to interfaces? Or are the interfaces maps to begin with, laying out the terrain before me? And what terrain do these interfaces describe, exactly (figure 15.4)?

The screen glows before me. I reach out to it. Am I touching another map, a mobile map in time, fleeing backward? If so, what was just here? What did I miss? Other maps. Menus pulling down laundry lists, texts, buttons, the colorful boxes and bubbles we press for their wiggle of acknowledgment. Icons, little actors on the desktop, waiting for stage directions. Operating systems do not only impose order but behavior: the desktop includes animated icons that catch a user's attention through scripted behaviors. This is a symbolic world, interfaces filled with metaphors, functioning much like signs in other media. Interfaces are sets of maps? Green for valleys, brown for mountains, and icons for roadside attractions . . . Maps are the original medium to use shorthand systems like icons to abstract space and time. Interfaces are certainly maps, graphic visualizations of the computer world, of the net . . . something that depicts something to do with: space, landscape, topology, topography.

The Internet as a topology is a dynamic and fluid terrain. Following work in the early 1960s, electrical engineer Paul Baran (1964) distributed a series of influential papers entitled "On Distributed Communications" while working at the RAND Corporation. These papers proposed detailed networked communications models that would protect the U.S. communications systems from enemy attack. Baran suggested that distributed networks, as seen in figure 15.5, are less vulnerable to attack than other network structures. Baran's ideas were especially powerful and continue to influence the structure of the Internet we have today. When introduced, the distributed network design countered paradigms of the day, suggesting that unreliable links in the network system could be almost as effective as reliable links if there were enough of them webbed together. The density distribution of the unreliable links could counteract node or link failures.

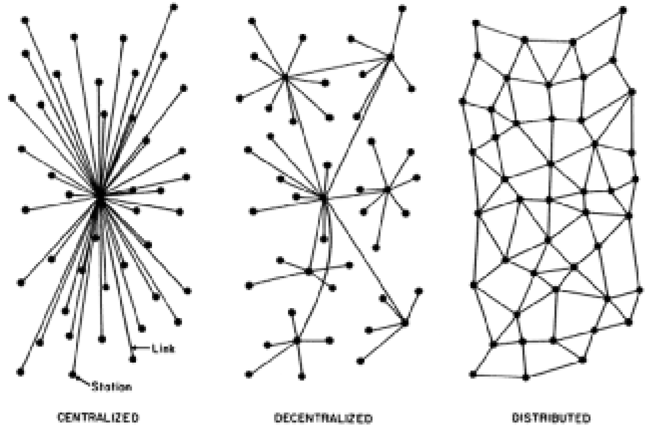
Not only does Baran's simple diagram of how the distributed communications systems function as a diagram or map of the structure of networked communications, but it offers the proper interface to understand them. Current interfaces use "back" and "forward," "pages" and "bookmarks"—metaphors still firmly rooted to the centuries-old technology of the book. Strangely, Internet browsers rarely show users how data gets from point A to point B, as though the structure of the system has nothing to do with its content. Rather, to see our maps form through alternate kinds of geographies might help demystify the topology of the Internet and of an individual computer.

```
/*Sebastian asks me, What is not a map? So many things can be called
a map.
*/
```

Currently, interfaces are abstractions that can be said to describe an underlying topology of the self. Interfaces become maps for our personal geographies with the computer, a user-centric geography instead of a spatially oriented one. What if we integrate these with other maps of the Net? Would differences between an individual computer and the network blur? This ever-present, fuzzy quality of networked computer culture is the third site at which to investigate interfaces.

the network

The men had come back to my apartment. There had been a very slight high-pitched buzz ringing throughout the building for two days, and they believed the source of the problem could be found within my apartment walls. They entered once again and began opening the holes they had covered last week, beginning a new kind of map, one of wires and strings, colors and connections.



| **Figure 15.5** |
Baran's network diagrams

Then they went to my neighbors,' an elderly man from Hong Kong, living in the other rear-facing apartment. His walls were covered in brown pegboard rimmed with S-hooks to hold cups and cooking implements. They started digging a hole from his side, pulling off his pegboard coordinates to connect our spaces, starting first with screwdrivers, then sledgehammers. Gradually our two spaces became one, as the wall broke down. The buzz remained. How had the wall held itself together? Our two apartments were distinct only in decor and the shadow where the wall had once been, the little buzz echoing across the wood. My neighbor sat calmly at his kitchen table, directly across from me as I typed, mirroring a part of me that must be tranquil, and I, perhaps, reflecting his interior agitation. I liked that we were a part of each other's lives today and might not be tomorrow. For once, looking at the stranger, I was not nostalgic or filled with longing, but perfectly in love with precarious balance. I smiled at him across our respective shiny kitchen tables.

The nature of instability is woven into the fabric of the world. As the universe expands, we chase behind it, part of the Copernican flock. We take along our plaster-smashed piles and boxes of papers and favorite mug. My velocity of recession is directly equal to your distance. Could my body swell, until I only feel my teeth and your nimble fingers? The penetrating quality of the wind depends on its ceaselessness—this is what makes the wind so powerful. And, time is its instrument, as it is in Hubble's formula. Is the outcome of all things disintegration and dispersion, or does a new beautiful order emerge from trajectories, breezes of attraction and repulsion, spinning magnificence for the while until the equation no longer works? I feel relief when my wireless network finds a signal. Somehow in the chaos around me, I am connected—and this may be *even more true once we move from a fixed network model to an ad hoc network model, where connections occur as needed.*

You are connected to me, I am connected to the neighbor, electronically. Somewhere. Our banks, our leases, our Amazon purchases, friendster and blog pages connecting us together along the speed of subscription. Reach. There is little there to see, but the social fabrics that look as if they are intangible are in fact made of sturdy threads, carrying weight in the semblance of tinted text and elusive images. How do we get from here to there in a space that needs no x, y, z-coordinates to exist, when we cannot easily define the structures upon which our content rests?

The overlap of the physical world and the virtual is a permeable and indistinct site of exchange, with certain nonmixable elements—for the time being: the sky, coffee, touching. This line, unlike the chemical properties of oil and water, shifts through time, changes with the latest buzzwords, technologies, home-grown wikis and party e-vite sites. And obviously there is cause for concern because all of this data can and will be monitored by disciplines of power: state and corporate interests.

The rapid spread of Internet accessibility and everyday use closely mirrors other patterns in the spread of twentieth-century domestic technologies. For the first time since the advent of the telephone, the space of the house has been dramatically redefined. A useful study by Carolyn Marvin (1990) details how the telephone was the first electric medium to enter the home and challenge traditional boundaries between public and private space. Likewise, the growing numbers of computers in the home means that many households are connected; it also signifies a shift in the way domestic space is understood. The Internet is a primary communication paradigm for the majority of people in the United States. It is no wonder that the physical world and its associated tools are changing to reflect the way we think inside the network. From contact lists on telephones and chat tools to multiuser online games, computational thinking has infused our everyday interactions.

The software packages and operating systems surround us, and we crave more and more connectivity. Browser-specific tools such as search engines, for example, offer one of many conceptual frameworks for desire. Yet for the majority of computer users, the hidden mechanics of systems are a satisfactory, even expected, given. Like Leonardo da Vinci's encoded notebooks, we live in a culture in which we create machines to keep our secrets from us. Users must like secrets, for more and more we desire ease of use and effacement of the technical workings of daily life.

We can map these connections through games as one type of framework—maps have grids, construction sites have scaffolding, spiders have webs. Examining ubiquity according to geography is only one way to glimpse how everyday computing has changed us.

```
/*Sebastian asks me, Where does the map end?
*/
```

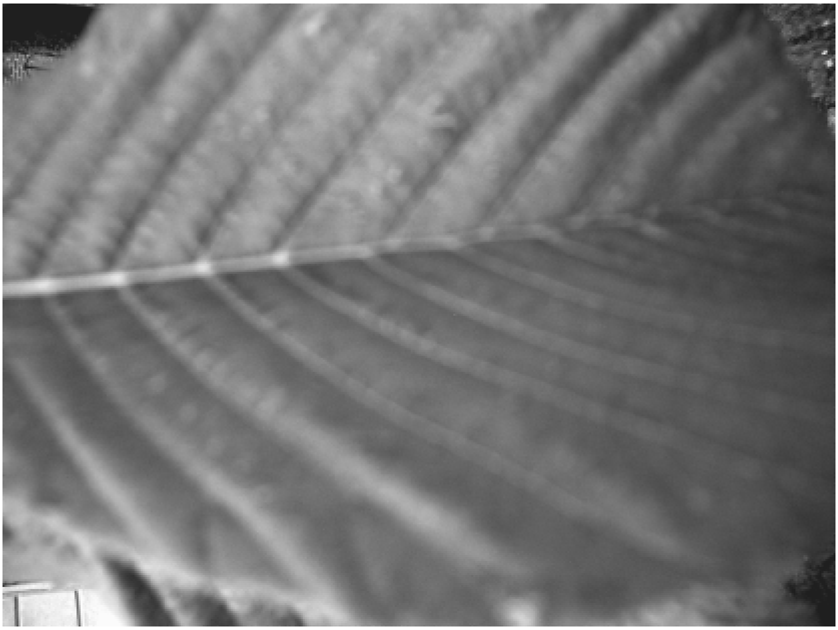
Mapping ubiquity. Interfaces are maps, graphic visualizations of the Net, depicting how software and hardware creators structure the terrain. The real and Baudrillard's hyperreal images have contaminated each other, or rather, merged, enacting a self where reflection and physical identity are one and where cultural forms and selves are reproduced endlessly. This site of blur happens most often with an individual's use of the computer. Images are never fixed, texts deleted and updated, leaving no trace of origin or time. The blurring of space, location, and time leaves no sense of near or far, past or present. As Foucault (1980) noted, ubiquitous disciplines of power control society through surveillance and control how the body is constituted. Many cybertheorists have argued that the network now extends the body through connections on the Net. And this body is itself extended, ballooning like a slippery tent through the world and even to such distant places as Mars. Skin is ubiquitous; it becomes a schematic of abstract spaces, Moscow and Berlin and La Paz and Caracas and Atlantis. Skin is a living record of time and dislocated place: a living, wrinkling map. We live through this skin and this map (figure 15.6).

The equivalent to skin and its markings lies in code, in programming. Computer programming provides the ultimate map, for it is both a language with its symbolic representations, and itself a body, a place where language transcends representation and becomes action.

objects

If interfaces and systems are written in code, are they ultimately linguistic maps? Must a map be a nonlinguistic representation of a space? If not, code, created through programming, provides the ultimate map. When we look at surfaces, or look inside, we see that maps contain the mapmakers and their values—the viewpoint of the mapmaker makes the map. And when programming languages, the only human-designed/created languages, are used, do they shape the map by the thoughts and biases of the makers and implementers of the programs?

Jef Raskin (2000), the inventor behind the Macintosh computer, noted that we need systems that better reflect the way humans work. He argued that software packages with separate types of interfaces do not meet user's needs first, and yet they structure how we conceptually think of tasks. The metaphors we use—desktops, navigation, "going here," "searching," are the models behind many software interfaces, molding how users understand digital experiences.



| **Figure 15.6** |
Natural network lines

Computer programming is the heart of the creation of software, and it too is influenced by metaphors—in fact, metaphors in programming have a strong affect on how digital tools are made and used. Therefore, a look at the structures informing the programming process—specifically at the way objects are created in object-oriented programming—is essential in the exploration of the way in which everyday digital experiences are also structured. I also look at mathematics, because some of programming—though not all—is grounded in the mathematics of doing: numbers, functions, and algorithms that one can conceivably construct.

```
/*Sebastian asks me, Are you interested in semantics or in language
structures, ways of saying or ways of thinking?
*/
```

When studying programming languages, we distinguish between semantics and syntax. While we can program in an object-oriented style in a particular programming language (e.g., C++ or Java), the design of object-oriented style is a separate entity. The idea behind the words we use to create an object—say, a chair—are one thing that represents the ideal state of the actual; the way we call a chair or multiple chairs into being, instantiating them, is another. Under this platonic nature of coding, the class is the ideal form, and the multiple chairs generated from the class instances of the ideal.

In technical circles, object-oriented programming is the most commonly applied programming approach, used for programming everything from databases to games. Object-oriented programming is both a procedure and a metaphor. When writing object-oriented software, programmers define the world in terms of a set of objects. A crowd would be, for example, one hundred instances of person objects, with various property changes (size, shape, color). Object-oriented programs are typically structured in a hierarchy of objects, with sets of objects or individuals having particular behaviors. In object thinking, an object is a Boolean shape, like the human body: inside an object is true, outside an object is false, and the surface, the skin, must be defined if it is to be recognized as another object. Objects are distinct, malleable, and controllable. They can be programmed to make choices and to behave in particular ways. They have their own properties and can encounter the properties of other objects at various hierarchical levels.

Whatever form one's conception of mathematical reality might take, such systems cannot escape their cultural background. Acknowledging that truth, objectivity, scope, and scale in mathematics are concepts based on a particular culture at a particular time, mathematician Raymond Wilder (1981) argues that disciplines affected by mathematics are just as influenced by the culture at large as by any other discipline, whether it be art, music, or scientific fields such as medicine. Mathematical logic diffuses from mathematics to the natural sciences and technology, infusing these related disciplines with methods and concepts. Object thinking has certainly infused both the culture of computing and the technologies developed by the industry. Contemporary interfaces, for example, map not so much spatial geographies but objects. Functions are broken down into one-word commands, and applications that may complete many kinds of related activities are represented under one icon: one product per icon, one icon per site on the map. Like scenic spots or roadside rest areas, icons invoke a spatial phenomenon through the differentiation of a place as an object. The distinctions and delineations between documents and programs, even the desktop metaphor with its objects in hierarchies, seems to follow the separateness of object-oriented programming. What object-oriented programming fails to deal with well is the fuzziness of boundaries and borders, when something needs to cross different kinds of object models, or when things operating in a given system are contradictory.

Advocates of object-oriented design argue that it represents a "natural" way to think about the world, even when programmers new to object-oriented design have difficulty identifying classes, or groups/types of objects, and forming hierarchical relationships among objects. In fact, it may take more than one semester to learn object orientation properly. While more efficient and reusable as a coding paradigm, the object-oriented model assumes that the world is made of objects and relationships to those objects, and this in turn shapes a way of seeing the world accordingly. I would argue that object thinking is not necessarily a "natural" way to think, but rather one of several kinds of epistemological practices that are not consciously recognized as ideological by program designers and programming practitioners. The object model, for one, reinforces a rationalistic and deterministic view of problems and solutions, creating separations and hierarchies between and among discrete objects. This could be because of the way computer programming/system design disciplines have been institutionalized as an engineering or scientific field, rather than, for example, a creative arts field, which it most certainly can be.

Gilles Deleuze (1993), in his investigation of creative trends in "nomad art," offers thoughts to use to reflect upon programming as a language, as an art form, and as a place. He remarks that one must be immersed in the material of creativity at hand, and here—that is code. Then the creative practitioner must collapse the visual aspect, or mere observation in favor of losing oneself into "the landscape" at close range. In an era that is characterized by the use of computers for a multitude of functions—one tool is used in disparate and variegated fields by a multitude of diverse users—the material or medium of everyday work and play is the computer. According to Deleuze, "The haptic function and close vision presuppose the smooth, which has no background, plane, or contour, but rather changes in direction and local linkages between parts" (1993, 169). Here, not only does Deleuze's description echo those of the Internet's structure, but it also recognizes the idea of the interface: the necessary haptic function, providing for close vision. To imagine this nomad art with a computer means losing oneself to the inherent properties of code and the possibilities of code: another kind of interface, not graphic, somewhat linguistic, primarily metaphoric and structural. From the box to maps to networks to programming, the metaphors and structures that come along with such designs must be continually sampled and shaped since they have the power to effect, systemically, mundane computer artifacts. New metaphors will arise—and they will be most interesting, and most socially relevant, from an infusion of new thinking and new authors among computer system and programming design. This means recognizing the importance of code and the structures of computational and algorithmic thinking, paying attention to the things that pass between organisms and systems, and creating new maps, structures, and computing paradigms along the way.

Work Cited

Baran, Paul. 1964. "On Distributed Communications." RAND Corporation. http://www.rand.org/publications/RM/RM3420/RM3420.chapter1.html (accessed April 15, 2004).

Baudrillard, Jean. 1994. *The Illusion of the End.* Stanford, CA: Stanford University Press.

Bush, Vannevar. 1945. "As We May Think. *The Atlantic Monthly* 176, no. 1 (July): 101–108.

Deleuze, Gilles. 1993. *The Deleuze Reader.* Trans. Constantine V. Boundas. New York: Columbia University Press.

Foucault, Michael. 1980. *Power/Knowledge.* New York: Pantheon.

Greenwald, A. G., and M. R. Banaji. 1975. "Implicit Social Cognition: Attitudes, Self-esteem and Cognition." *Psychological Review* 102: 4–27.

Kuhn, Thomas S. [1962] 1996. *The Structure of Scientific Revolutions.* Chicago: University of Chicago Press.

Marvin, Carolyn. 1990. *When Old Technologies Were New.* New York: Oxford University Press.

National Science Foundation. 2000. "Women, Minorities and Persons with Disabilities in Science and Engineering: 2000." http://www.nsf.gov/sbe/srs/nsf00327/access/exec.htm (accessed April 2004).

Norman, Donald. 2002. *The Design of Everyday Things.* New York: Basic Books.

Phelps, E. A. 2001. "Faces and Races in the Brain." *Nature Neuroscience* 4: 775–776.

Phelps, E. A., K. J. O'Connor, W. A. Cunningham, E. S. Funayama, J. C. Gatenby, J. C. Gore, and M. R. Banaji. 2000. "Performance on Indirect Measures of Race Evaluation Predicts Amygdala Activation." *Cognitive Neuroscience* 12(5): 729–738.

Raskin, Jef. 2000. *The Humane Interface: New Directions for Designing Interactive Systems.* Reading, MA: Addison-Wesley.

U.S. Department of Commerce, Economics and Statistics Administration. 2001. "A Nation Online: How Americans are Expanding Their Use of the Internet." http://www.esa.doc.gov/anationchart.cfm (accessed November 10, 2003).

Wilder, Raymond L. 1981. *Mathematics as a Cultural System.* New York: Pergamon Press.