Ted Selker, Guest Editor

New Paradigms *for* Computing

As researchers and product designers, we are colleagues as well as competitors, creating systems that will support people in their work. But striking new paradigms, styles of working that could shape the whole field, are usually "in the air" long before they mature and become accepted. We are often working on parts of what will turn out to be a shared vision. A wonderful new tool is envisioned: some try to create it, some find it not useful, and some improve it. The cycle repeats.

How do we as an informal community of innovators choose our projects? Which paradigms for using computers succeed? How do we nurture emerging paradigms before they are ready for widespread acceptance?

Over the last three years I have had the honor of hosting a workshop at IBM's Almaden Research Center entitled "New Paradigms For Using Computers" that addresses these issues. For these meetings, we have invited industry pioneers whose efforts have created many of the new ways we use and think about computers. These professionals, and many others, have been the trailblazers who have created an industry and its ability to change the way people do things. In this small set of articles, we really can't do justice to this topic, but instead hope to offer glimpses into the creation of new ways of using computers.

We begin this section with a short commentary from Xerox PARC's vice president John Seely Brown, a pioneer in AI and education who nurtures and is guru to Xerox PARC—the everpresent foundry of new paradigms in computing interfaces. Brown's piece should not leave us with the feeling that technology advancement is strictly evolutionary. While he cautions us that it is hard to visualize too far from what exists, I am happy for the non-incremental approach and legacy from the actual work that Brown's PARC represents.

MATSU

Nolan Bushnell, the founder of Atari who is often credited as the creator of the computer game industry, explores the "Relationships between Fun and the Computer Business." In his article he reminisces, assesses, and projects the effects from the game industry to the computer industry as a whole. Bushnell's article makes tangible some of the incredible distance we've come since the 1970s—from a world where manufacturers didn't take cathode-ray tube use in the computer industry seriously to a time when most two-year-old children know about computers. Bushnell's way is to mix lessons learned from the game industry with some as-yet unexplored visions of the way we will play.

Henry Lieberman's work has been that of creating programming and idea presentation environments using his favorite tools of creation objects (his work on actors), dynamic languages (his work on garbage collection), and interactive environments (his work on programming by example). Lieberman's article, "Intelligent Graphics," focuses on the inventive serendipity that has and can come out of the relationship between AI and graphics—an idea promoted by yet another pioneer, Muriel Cooper. He also briefly discusses programming by example, a style of interaction that he has been working to demonstrate and promote over the last couple of decades.

Ken Kahn also cares deeply about programming languages, having devoted his research career to creating Prolog interpreters, parallel logic systems, and logic-oriented systems. In his article, we see him engaged in his first love, that of creating visual approaches for thinking about programs. Watching programs execute, seeing their structure, seeing their function, organizing them—all with the underlying goal of making programming as easy as child's play.

Finally, as you look to my article, you see me not working or talking about any one idea as I could, but giving a review with some context that describes many of the ideas shaping the way people are using computers today.

I hope this eclectic collection of articles can serve as a celebration of our temptations to explore and design the ways computers will help us communicate, create, and record our experiences.

To Dream the Invisible Dream

When it is written, the history of computers will, I believe, be quite simple. In the beginning was the computer. Then it disappeared. Of course, it didn't go away completely. It just dissolved. Either it became part of the physical background, forming part of ordinary objects such as tables, chairs, walls, and desks. Or it became part of the social background, providing just another part of the context of work.

Indeed, this second phase of the history of computing is already under way. The modern car is really a four-wheel computational platform. Yet I'm rarely made aware of this when I drive it. Furthermore, when I go to the automotive showroom, I don't have to ask what operating system or presentation manager the car uses. Here, at least, computers have finally gotten out of the way.

The field of human-computer interaction is really configured around this central paradox. Designers struggle to produce simplicity out of complexity, direct connectivity out of mediation. Instead of drawing attention to itself, the best design lets us reach through computers into the world, allowing us to focus on creating value, not manipulating tools. So, for example, in panic stops and radical curves, the computational power in my car doesn't add to my problems by drawing attention to itself. Instead, it invisibly helps connect me to the road and the world outside.

From this perspective, I see the new paradigms for design and use developing hand-in-hand. As they adapt to current practice, new technologies become less visible. Yet, simultaneously, by adopting these new technologies, current practice continuously evolves.

Clearly, this is not a view of radical transformation. We all love to be radical and to pursue radically new ideas. But our experience shows that fundamentally new technologies seldom get adopted in a discontinuous fashion. When put to real use doing real work, new inventions almost always miss their mark, no matter how many tests ran in the lab.

Adaptation and adoption require extensive fine-tuning in the real world. The passage from the Lisa to the Mac is a famous example of this. At the same time, the mistakes Apple made in over-hyping the immediate potential of the Newton show how hard it is for any of us to learn this lesson. Nevertheless, I think we all need to learn it. Instead of focusing all our attention on radical transformation, we should try to understand the dynamics of "radical incrementalism." This is what turns radical invention into innovation.

—John Seely Brown