THE EDITORIAL VIEW

EASE OF USE: THE BIGGEST BARRIER

By Michael Slater {5/29/00-01}

Relentless advances in semiconductor technology continue to enable a dizzying array of digital devices that promise—or threaten—to pervade nearly every aspect of our lives. There are countless technology challenges to overcome, but the underlying Moore's Law trend and the massive amounts of capital being invested will enable the industry to jump most of these hurdles. The biggest question is not whether any given technology or infrastructure achievement is possible but whether the result will be products that are effective and enjoyable to use.

The technology industry's track record for designing products with high-quality designs—from the user's perspective—is, to put it bluntly, pathetic. Because of the benefits computers and other devices offer, we have come to accept the frustrations they cause. But it doesn't have to be this way. New types of information appliances and other digital devices will have a far better chance of success if their creators pay a lot more attention to usability.

Many books have been written on user interface design, but they do not seem to have made a great impact. The Macintosh user interface standards helped drive an important wave of advances, but they also established a mode of design that has become limiting. Furthermore, the new world of information appliances demands a different approach than does PC software.

Information Appliances and Beyond, a new book edited by Eric Bergman, provides a valuable collection of case histories written by appliance designers—but it stops short of providing a prescription for design. Another new book, *The Inmates are Running the Asylum* by Alan Cooper (*www. cooper.com*), provides great insights into why most designs are so bad—and, more important, how to do things differently. Unfortunately, huge practical hurdles still block the path to implementing many of Cooper's recommendations.

At the heart of the problem are the people who do most user interface design: programmers. This is not to belittle their intelligence or their intent; it is simply to point out that they are fundamentally unsuited to the task. They tend to be people who have an easy time managing many details and remembering complex processes—a trait not shared by the majority of the population. Their thinking is tremendously influenced by their intimate knowledge of the internal structure of their programs, and they tend to create user interfaces that mirror these structures. Lots of people talk about creating user-friendly software, but few programs are actually designed well enough to meet the user's goals.

Many programs are further crippled by interface designs that are created after the fact: first the program is written, then an interface is put on top of it. The interface should not be just a pretty face; in fact, "interaction design" is a better term than "user interface," because it emphasizes that what is important is not just the appearance of the interface but what it does. When professional designers are involved, all too often they just put a pretty face on a lousy interface—what Cooper calls "painting the corpse."

Programmers have an irrepressible tendency to design programs for people like themselves, and they are also overly influenced by ease of implementation. It is natural for them to dismiss as "not worth it" any user interface enhancement that increases the coding effort. Someone other than a member of the programming team should specify and enforce the interaction design so it doesn't get compromised by the programmers, whose priorities simply don't match those of the users.

Usability testing done after a design is complete may provide some useful tweaks, but once the code is written, fundamental changes in the interaction design are rarely given serious consideration. Great usability can come only from designs that are superior from their outset, not from patched but fundamentally flawed approaches.

Dramatically better user interfaces are possible if two key changes are made to the design process: the interactions must be designed before the program is written, and they must be designed by people trained in the art of interaction design and with a deep appreciation of the user's needs and desires.

Unfortunately, both changes are hard to put into practice. There isn't a pool of designers nearly big enough to create interfaces for the number of products in development. Few people see interaction design as a career path, and precious little training is available. Industry should work with academia to dramatically increase the number of people going into this area of design, and the industry should value and compensate them highly to make the profession attractive.

Even if people with the required skills are available, time-to-market demands make it difficult to provide them with the time they need to do their work. More than ever, in this era of "Internet time," there are enormous pressures to get products to market as quickly as possible. It is hard to argue for a three-month (or longer) extension of a development schedule to provide adequate time for a first-rate interaction design. The prevailing attitude is to get the first product out as quickly as possible and fix it later. By that time, however, needed structural changes may be hard to make, both technically and emotionally, and may be disruptive to those early adopters who have managed to learn the initial product.

Bad design is not limited to high-tech products, but the internal complexity of these devices makes good design especially important. We are at risk of creating a new wave of information appliances that give us new capabilities but carry a high price in frustration. Many products will fail, not because the product concept is bad but because the interaction design is faulty. There are no magic bullets, but with the proper effort the industry can do dramatically better than most existing products. ◇