

# May the Best Interface Win!

**A**MAZON.COM MAY NOT BE the model of financial prosperity, but its Web site is one of the Internet's best shopping locales. By making its online store easy to navigate and use, Amazon has created an environment where money is easy to spend. The company may still be months—or years—from significant profitability, but that didn't stop its customers from buying more than \$2.7 billion worth of goods last year.

With all that money pumping through Amazon's site, you'd think that other online retailers would take notice. Not true! Most companies still have tremendous difficulty building

Web sites that are easy to use—despite years and billions of dollars spent trying. Considering the money that stands to be gained, you would expect that more companies would be following Amazon's lead.

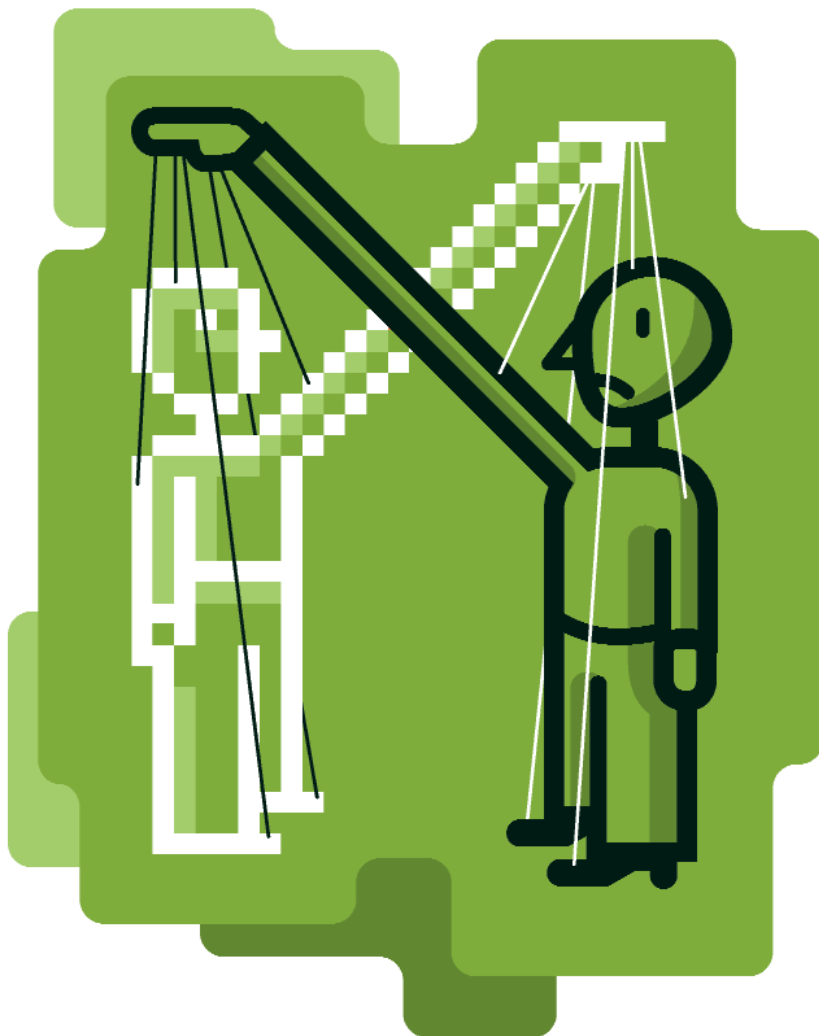
Take something as simple as electronic shopping carts. Last fall, when Amazon changed its privacy policy to claim ownership of its customers' personal information, I decided to protest by giving a competitor, CDNow, a try. I was impressed by how CDNow let me search for albums by artist, record label, and album and song title, and I liked being able to hear 30-second samples of the songs over the Web

(even though the sound quality was pretty poor). On the other hand, CDNow's site was glacially slow; even with my cable modem, it took me 45 minutes to pick out five albums and put them in my shopping cart.

So far, so good. But before I could whip out my credit card and hit the "buy" button, something came up and I had to get up from my computer. The moment had passed. A few days later I remembered that I had never bought my CDs, so I surfed back into the CDNow site. But when I clicked on the cute shopping cart icon, it was empty—my cart had expired. Rather than go back tediously through the CDNow interface, I went over to Amazon, typed in the names of the albums, and then headed for the checkout. Besides the \$75 worth of CDs, my shopping cart also contained \$150 worth of books that I had picked out a few weeks before but neglected to order. Unlike CDNow, Amazon doesn't throw out the contents of your shopping cart, even if months have passed since you made your selections. This may sound like a trivial detail, but by paying attention to shopping usability, Amazon kept this customer happy—and tripled a sale.

The computer programmer in me says that maybe shopping carts *should* expire—it's a waste of disk space to save data for millions of shopping carts that might never be used again. In a real supermarket, after all, clerks take the food out of abandoned carts and put it back on the shelves. But Web sites are different from physical stores, and disk drives are incredibly cheap these days. For a few hundred dollars, CDNow could buy a disk large enough to store more than a million electronic shopping carts. The company's decision to empty mine cost not only that one sale, but future sales as well.

Along the same lines, consider another dubious, but common, online practice: forcing Web shoppers to type



credit-card numbers without spaces or dashes. An overwhelming number of sites demand that visitors key in a continuous string of 16 digits or more. No spaces or dashes please! This is so commonplace that hardly anyone notices its tremendous inconvenience.

Take a look at the front of one of your credit cards, and you'll see that

**Well-written software doesn't require that people bend their behavior to the machine; it responds productively to the quirks and foibles of its human users. E-commerce sites, take heed.**

the account number is printed in four groups of digits with spaces in between. Those spaces are there for a reason: they make the digits easier to read and reduce the chance that you'll get the numbers wrong when you type, speak or otherwise transcribe them. Yet how many times have you tried to use that card at a retail Web site and been forced to cram all the digits together into one box on the purchasing form? And how many times have you had to fill in the box—or even the whole form—a second time because you screwed up the credit-card number on the first go-round?

It's another one of those small details that annoy me as a consumer. This one infuriates me as a programmer, too, but for a different reason: I know that in most computer languages it takes precisely one line of code to remove the spaces from a field of data provided by a user.

Digital River, of Eden Prairie, MN, builds, hosts and manages the Web sites of more than 8,000 clients, including Novell and ScanSoft. The company bills itself as a global commerce service provider. So I called up Digital River and asked why its credit-card processing system doesn't usually let people type credit-card numbers the way that the numbers appear on the embossed plastic.

"Because that's the way we wrote it!" said Marty Boos, Digital River's vice president of information systems. "It's never been an issue for us. We haven't gotten complaints on it."

One reason that Digital River may not get complaints, I explained to Boos, is that the company puts a little note on its Web pages that says "Do NOT use spaces or dashes" when entering credit-card numbers. So clearly the fact that Digital River's computers don't take spaces or dashes has been an issue for the company. Except

that instead of reprogramming its computers, Digital River decided to reprogram all of the human beings who use its e-commerce sites.

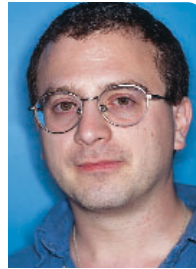
Boos didn't see the irony. Instead, he agreed with my assessment. "When given instructions, 99 percent of the customers can [follow them]," he said proudly.

I find this attitude distasteful. Well-written software doesn't require that people bend their behavior to the fancy of the machine; it responds productively to the quirks and foibles of its human users. One nice feature of Microsoft Word, for instance, is that if you type a word with a common misspelling, the program will automatically correct it. Fixing typos, letting people type spaces in credit-card numbers and even letting people type the letters "O" and "I" for the numbers "0" and "1" isn't rocket science.

Unfortunately, most of the universities turning out the next generation of computer programmers aren't spending much time teaching students the art of writing humane, usable software. Undergraduate computer science departments have instead crowded their curricula with the harder aspects of the subject, like data structures and compiler design. I searched for the word "usability" on the Web sites of the computer science departments at Stanford University, MIT and the University of Pennsylvania and came up with nothing. Searches at Harvard University, the University of

California, Berkeley, and the University of Washington were slightly more fruitful: each turned up a single homework assignment for a single advanced course that mentioned usability issues.

To be fair, Web searches like this can be misleading. Daniel Jackson, who teaches MIT's undergraduate software-engineering course, says that he



tries to introduce the topic of usability from time to time. But it's hard. That's because, Jackson says, before a programmer can create a usable computer system—whether it is a desktop application or an e-commerce Web site—he or she must have a clear conceptual mastery of the task at hand.

For many programmers, that is no fun at all—and as a result it frequently gets omitted. "It's much harder to analyze a problem deeply than it is to hack a bunch of code," says Jackson. "Too many software developers don't care about the user enough to care about what problem they are trying to solve. They get too tied up in the joys of their own solutions."

Going back to electronic shopping carts, it's fun to think of clever ways of "garbage collecting" abandoned carts and freeing up space inside CDNow's database server. It's no fun at all to calculate how much space is being wasted on Amazon's servers and how often the company needs to buy new disks and faster database servers to keep up with all of the dead—or dormant—data. Likewise, it's no fun thinking of all the ways that somebody can mistype a credit-card number and trying to make an e-commerce system tolerant of user error.

It would be nice if the financial markets would reward the companies with the best interfaces, but there are so many other factors determining a company's success or failure that usability invariably plays second fiddle to issues like marketing and financial acumen.

But me, I'm rooting for Amazon—and hoping the best interface wins. ■