Looking Forward Can Hurt Your Brain

To the Editor:

I’ve got to admit, your January Outlook issue got me thinking. The spatial conferencing described in “Wearable Devices: New Ways to Manage Information” (Mark Billinghurst and Thad Starner, pp. 57-64) was very interesting. Those faces you see in the figures must be facing a camera, right? How do they see your face? Hmm. What if there was a device you could wear that could sense the shape of your head—your face, eyes, and everything—and transmit it over the Net? Maybe by running a slight current through your body, who knows?

The EveryBook in “As We May Read: The Reading Appliance Revolution” (Bill N. Schilit et al., pp. 65-73), with its two letter-sized screens, was also intriguing. What if we replaced the keyboard on laptops with a touch-sensitive screen? What if the computer had text-recognition built in? What if it had the Mac’s ability to move items from screen to screen? What if it had two cameras so that it could capture a 3D image of you?

And what if you were driving along and got hungry (Ted Lewis and Benjamin C. Fuller, “Fast-Lane Browsers Put the Web on Wheels,” pp. 141-144)? What if the computer could give you restaurant locations and have your meal waiting for you?

Okay, that’s enough from me. My brain hurts.

Gary Kephart
gary_kephart@pobox.com
http://www.pobox.com/~gary_kephart

ALONE ON THE ROAD LESS TRAVELED

To the Editor:

I enjoyed the “Fast-Lane Browsers Put the Web on Wheels” (Ted Lewis and Benjamin C. Fuller, Jan. 1999, pp. 141-144). However, two issues struck a chord.

Lewis speculates that “50 billion wasted hours” are caused by commuting. My commute is a half hour. For the most part, I enjoy the chance to formulate my thoughts in the morning and wind down in the evening. This is not wasted time for me. I also find that the time away from phones, browsers, and the like definitely contributes to the quality of my life. (Note I do not have to commute on crammed roads either. The road less traveled is better!)

I also encourage the wise use of technology. There may be no oil crisis today, but we should still be doing all we can to reduce our dependence on fossil fuel. Providing Web services and other conveniences in the car only increases our dependence. Of course, maybe the vehicle was electric, but then where did that energy come from?

David J. Pfaltzgraff
Senior Software Engineer
Patapsco Designs, Inc.
Frederick, Md.
davepfz@fred.net

NO SHORTAGE OF QUALIFIED FACULTY

To the Editor:

If Computer is to serve all members and not be just a shill for academia, then the editors need to vet the wild claims such as those contained in “Employment in 1999: Opportunities Amid Challenges,” (John Edwards, Jan. 1999, pp. 19-22).

I submit that there is no shortage of qualified faculty for computer science, or engineering, or any other field. There is a shortage of fully qualified persons, including those with PhDs, who are willing to work part time at below-market rates with no benefits and no chance for advancement or tenure. Full-time, tenured faculty make big bucks. Their expectation seems to be that others should work for a pittance while they dip their ladies into the gravy bucket. They epitomize chutzpah.

My first department head had only a master’s degree. Even though he was well known in the field and had written several textbooks, how many schools today would consider him for anything more than a temporary part-time instructor position? Instead, he could easily choose to work full time as a consultant. Why should he work for below-market rates in a position beneath his proven skills?

When schools stop overspecifying qualifications and start paying market rates, they will have a surfeit of faculty applicants. The free market works, and the whining from academics won’t by itself create applicants.

William B. Adams
Springfield, Va.
william.adams@ieee.org

WHAT’S NEW ABOUT COTS HEADACHES?

To the Editor:

I was greatly disappointed with “COTS Integration: Plug and Pray?” Continued on page 6
Letters

Continued from page 4

(Barry Boehm and Chris Abls, Jan. 1999, pp. 135–138). Having been involved in COTS integration projects, I am acutely aware of the pitfalls. I was hoping to see more encouraging advice. Instead, I found old ideas and marketing for a new product. How very sad it is to see such a meager offering from one of the leaders of our industry.

But my greater disappointment came in the realization that our industry seems to be on a questionable course. A dozen years ago, we were pursuing software reuse as a means of shortening development time and increasing reliability. The idea was to construct new applications from reusable components. Components would be designed specifically for reuse: they would be thoroughly tested and documented; they would be flexible and adaptable at the time of use.

Now it seems that we are attempting to use COTS products as reusable components. This is inherently costlier (from a maintenance perspective, as the authors point out) and riskier. COTS products offer none of the advantages of the ideal reusable components we once sought. By and large, they are not meant to be used as part of a larger system; many are written to be the centerpiece of the system, if not the system itself. The documentation for these products frequently falls victim to the ship date, as does the testing. Change is not controlled and managed internally. Rather, it is dictated by external market forces, as the authors note. Moreover, COTS products are provided without warranty, and modifying them internally usually assures the lowest level of vendor support. COTS may offer an expedient alternative to developing a reuse repository, but we are kidding ourselves if we think we are reaping the expected benefits of reuse.

Just because the current trend is to use COTS products to assemble a system does not mean it is the appropriate way to proceed. Developing tools to support the effort may help us to do the job better, but I believe we should not be doing the job at all. Since Boehm and Abls did not ask this question, I will: Are we, as an industry, on the right track with COTS or will we regret our path five or 10 years from now? I invite the comments of my peers.

Stephen Blanchette Jr.
stephen.blanchette@acm.org

Barry Boehm responds:

Because I wrote a column for managers, as Computer requested, it is not too surprising that technical experts such as Blanchette with extensive hands-on integration experience would find the column’s ideas to be old. However, his letter supports the column’s motivation: Managers are indeed attempting to treat COTS products as reusable components, often with adverse effects. The fact that managers continue to be surprised by these adverse effects indicates that for many people, the material in the column may be new and useful.

I disagree with just one statement: that COTS products are inherently costlier from a maintenance perspective. We tried to make careful distinctions about when people would find COTS solutions more expensive to maintain (for example, when the application is relatively stable and the COTS products are relatively volatile).

Indeed, COTS solutions are more often cheaper to maintain, since the main alternatives are to use unsupported componentry (a big maintenance risk) or to build and maintain the component yourself. I made the latter mistake in 1981, and I’ve tried not to repeat it. At that time, while at TRW, I helped survey all the available COTS forms-management systems. When it was completed in 1982, we had built what was arguably the world’s best forms-management system—and a big maintenance headache. TRW decided to scrap it in favor of a COTS product so it could share the maintenance costs (and the upgrades) with lots of other users.

In general, I think it’s folly not to use COTS at all. If we all become more informed COTS buyers, however, eventually market forces will drive out the poor COTS performers.

Stephen Blanchette Jr.
stephen.blanchette@acm.org