

E Pluribus Unum

Once, more than ten years ago, I voted for a Republican. I kept this from my father, who had taught me, when I was a child, that “Nixon” was a bad word. In fact, I tried to keep my Republican vote a secret from everyone. I felt really awkward.

Bear with me on this—there will be a point. It is the same point that I made in my initial essay as the Viewpoints department editor: that too often we compete when we should collaborate. I want to belabor that point, but first you get some Massachusetts history.

John Silber, a Democrat, symbolized many trends I have found troublesome and counterproductive in higher education. I therefore could not bear to vote for him for governor of Massachusetts. Better in this one case, I thought, to vote for the Republican—especially since this was a likable, good-guy, take-on-Nixon Republican—than to do anything that implied, if only to myself, that I approved of the Silber approach. I convinced myself that I was doing the right thing. Thus I and a lot of people like me in the People’s Republic of Massachusetts made Bill Weld its governor.

In retrospect, I was wrong. Silber did and still does stand against much of what I value in higher education. But Republicans are Republicans, and although Weld was a pretty good one, he nevertheless took the Commonwealth in very Republican directions. I did not like the result.

I should have known better. As a friend put it at the time, I should have held my nose and voted for Silber. When I faced a similar dilemma here in Illinois,

five years ago, I voted for some guy named Poshard—about whom I still can tell you almost nothing except that he is a Democrat. The good-guy Republican, who won the election, turned out, despite many virtues, to employ myriad staff who believed that selling driver’s licenses to truckers was a good way to raise campaign funds.

row and general progress over the longer haul. The latter often requires compromising, postponing, or abandoning the former. To make things worse, our vendors face a similar dilemma: they choose between making their products robust, enduring, and maintenance-free—one might say “of high quality”—and making them profitable. The two choices conflict because profit, these days, often lies in consultation, maintenance, and obsolescence rather than in initial sales.

Colleges and universities have rather similar information technology environments, a point I have made elsewhere while criticizing wired-campus ratings. We in higher education should therefore have relatively homogeneous needs, at least within sectors. Moreover, except in a few isolated cases, we do not compete with one another on the basis of information technology. We should thus be able to marshal considerable collective buying power, and with that power, we should be able to drive good deals with vendors.

But each institution prioritizes its needs slightly differently; in micro-economic terms, initial utility functions vary. Each is trying to avoid a different Silber. In the process, each chooses a different Weld. In order to slay our particular dragons, we in higher education approach vendors with different priorities. Vendors, most of which are not run by fools, take our expressed needs at face value. They negotiate separately with each of us, and most of us come out of those negotiations feeling pretty good. We have done well, as separate institu-

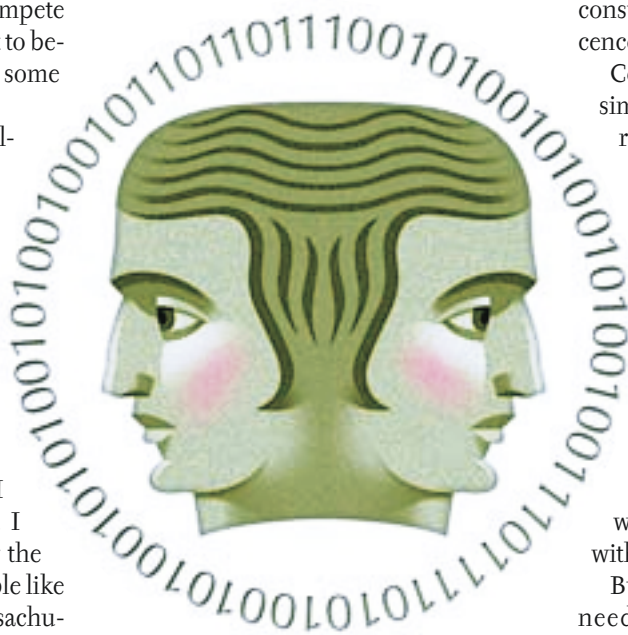


Illustration by Steve McCracken, © 2004

It is hard to balance narrow optimality and broad optimality. Voting for Weld, I sought narrow optimality: not electing Silber. In the process, I sacrificed broad optimality: moving Massachusetts closer to my political preferences.

We who “do” information technology in higher education have the same difficulty I had with Silber and Weld: we choose between specific goals for tomor-

tions. The problem is that separately optimal outcomes may not be collectively optimal; indeed, they usually are not.

I digress to an example. Network security is easy to attain for a personal computer: one grasps the clip on the network cable and disconnects it from the computer (or, perhaps, removes or disables the computer's wireless networking). A computer thus disconnected has high security. But now the user cannot read e-mail or browse the Web: the computer has low functionality. There is a fundamental and unavoidable trade-off between security and functionality.

Colleges and universities approach this trade-off differently. University A, for example, might decide that functionality is most important, that security is a problem for users but should not affect them unless they cause problems for others. University O, in contrast, might decide that users have an obligation to seek security even before problems occur and thus might insist on standards for configuring and authenticating devices before users gain access to the network.

Now consider the dilemma facing two companies—call them M and S. Both companies seek, in general terms, to provide functional, secure products. Because of the trade-off, however, each must decide whether to engineer its products toward functionality or security. M chooses functionality. It argues that users can interact easily with data and one another, since networking makes sharing simple and convenient, and that users can set up applications without complicated processes or specialized knowledge. S chooses security. It argues that securing applications and data from the start eliminates the costs of cleaning up security breaches later, more than offsetting initial setup complications and costs.

Each company believes that its choice minimizes support costs for its customers. Each markets in terms of its preferred definition: M trumpets functionality and is silent on security; S does the opposite. By defining their differences along different dimensions, M and S complicate choice for their customers—unless those customers happen to have corresponding utility functions.

What happens? University A finds Company M's pitch appealing, and Uni-

versity O likes Company S. They buy accordingly. M and S have happy customers. Meanwhile, another university, L, has not decided where it stands on the functionality/security trade-off. It seeks guidance from M and S, without success. Its peers A and O give divergent advice. Eventually L makes a choice but remains unsure it has done the right thing.

By acting separately with unrecognized differences, Universities A and O divide their buying power. They lose collective clout. And by not agreeing on open standards for the important functionality/security trade-off, Companies M and S waste resources on duplication—resources that could have gone to innovation. That is, A and O have encouraged M and S to compete in ways that neither serve the collective good nor promote innovation. Meanwhile, L is frustrated. It is making choices somewhat randomly.

We cannot do much about the short-term profit goals that drive both M and S to eschew reliable, robust hardware and software in favor of lower-quality goods for which they can sell maintenance and support. But colleges and universities

should certainly be trying to make things better. And this is where I return to my point of two years ago. If we in higher education were to use our collective buying power with determination and focus, we could have important influence on the market. This clearly can work: think, for example, about Internet2 and National LambdaRail, the Microsoft Campus Agreement, Eudora, Netscape, Kerberos, and the various educational discounts.

So long as those of us in higher education fail to resolve our unimportant differences before engaging vendors, Universities A and O will continue to pay more than they should, University L will remain confused, and all of our resources will be wasted. We need to figure out where our interests align, and we need to make the choices and compromises that will enable us to work together.

Gregory A. Jackson is Vice-President and Chief Information Officer at the University of Chicago. This is the final column in his two-year term as the Viewpoints department editor for *EDUCAUSE Review*.

