"Any clod can have the facts, but having opinions is an art."
Charles McCabe, *San Francisco Chronicle*

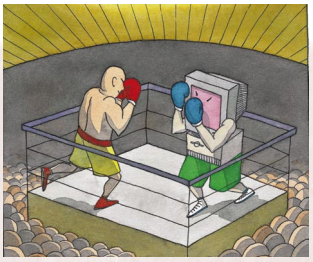
Mitigating Microsoft with Virtual Consoles

Neville Holmes, University of Tasmania

The admirable and usually prescient Ted Lewis may have got it partly wrong about Wintel when he guessed that "Microsoft and Intel are guaranteed a long and prosperous future with little intervention from [the US Department of Justice]" ("Who's Afraid of Wintel?" *Computer*, Jan. 1998, pp. 149-152). By now it's stale news that DOJ, through Joel Klein, head of DOJ's antitrust division, has moved against Microsoft.

However, a rereading of Lewis' well-argued column convinces me that DOJ's case against Microsoft is—at the very least—highly questionable. Either way the action won't be cheap for the US taxpayer. And Microsoft winning the battle could strengthen their position, which, of course, is not what DOJ's action is designed to achieve.

Even if DOJ wins, the effect could be bad. Writer Richard Miniter sees a danger that "Klein's efforts to save Microsoft's competitors may kill them" ("Busting Microsoft May Give Australia Free Rein," *The Australian Financial Review*, May 21, 1998, p. 21). Miniter also claims that "thanks to Klein, American corporations might not be free to compete in the global market; they could be at the feet of bureaucratic regulators who construe every innovation as a step toward monopoly."



If having windows is at the heart of what turns users on these days, then what we need is a separation of the windows from the software that runs in them.

Other commentators have made the same point, though rather less extravagantly. In any case, it looks like war. In war, the main casualties are all too often the bystanders.

Will Tracz, Lockheed Martin Federal Systems, MD 0210, Owego, NY 13827-3998; will.tracz@lmco.com

MICROSOFT'S STRENGTH

It is Lewis' view that Microsoft and Intel, left alone, will by 2017 "look as out-of-date and unimportant as Standard Oil is today." That may be true, but I think the process of obsolescence should be hurried up. As a disgruntled user of Microsoft's OS, I would like some changes, and I would like them soon.

My machine must be rebooted far too often. Furthermore, as a user who finds the keyboard to be quicker than the mouse, I am fed up with the increasingly inconsistent and decreasingly frequent provision of keyboard alternatives to mouse manipulations. I'm also fed up with the uselessness of Microsoft's various help facilities. Let's face it, Microsoft's software quality is poor.

Is the software competing with Microsoft's any better? I don't know. I just use what I'm given, and I'm given Microsoft. That's common today, and that's what Microsoft seems to have wrapped up. To get better software, Microsoft must be given some competition where it really matters: in the OS itself.

DOJ is trying to get at Microsoft from the applications side of the software structure in focusing on browser software, but that's the wrong side. Ted Lewis points out that "without its upgrade business from MS Office and Windows 95, Microsoft would be a mere shadow of itself." And as consulting engineer Bob Weeks makes plain in a letter to the editor (*Computer*, Feb. 1998, p. 4), Microsoft's dominance is due to official adoption of its OS. MS Office is strong because MS Windows is strong, rather than the other way around. In other words, Microsoft's strength is in its foundation, its OS.

The most striking aspect of computing in the 1990s, or at least the late 1990s, is the almost universal acceptance of windows as a way users interact with software. Users get a strong feeling of control, even exhilaration, in having several windows in production at once and being able to choose among them. The second most striking aspect in computing these days is the acceptance of second-rate software to run in those windows.

If having windows is at the heart of what turns users on these days, then what we need is a separation of the windows from the software that runs in them. Then simpler OSs and applications than Microsoft's could run in those windows.

VIRTUAL CONSOLES IN HARDWARE

Before windowing took over, the console—the display terminal together with the keyboard—was implemented primarily in hardware. Now that hardware is so much more capable than it used to

be, it could run multiple virtual consoles just as easily as it used to run single consoles. If we went to a hardware-based multiple-console model, the OS would be much simpler, and its programs would merely request a console from the hardware when needed.

On my desktop PC, the two windows I use most are a DOS window and a Unix window. It's just like having two machine consoles on my desk, but the windows make it much more convenient. The OS simulates two consoles, but I can place the virtual screens where I want and size them to taste. I can also easily shift between them (with Alt-Tab).

Windowing is, at least from the user's point of view, a straightforward facility. Each window is basically the visual part of a program's console, and when control is shifted to a particular window, the keyboard is temporarily the console's input device. The active program only gets to use its own console—or at least the keyboard or mouse—when the user shifts control to it.

With virtual consoles implemented in hardware, users would find it much easier to shift from OS vendor to OS vendor, and new OSs would be easier to develop without the burden of providing windowing support. Implementing consoles in hardware would let application developers compete with Microsoft where Microsoft is strongest: the OS. Furthermore, improvements in windowing could be developed in the hardware itself, probably with little need to change the programming interface.

Virtual terminals under control of hardware would also make it easier to support video applications and promote PC/TV convergence. And special-purpose OSs—to operate networked domestic appliances, for example—would be easier to develop and easier for people to learn to operate if we had a common hardware support system.

With Windows CE poised to expand into a number of markets, Microsoft could easily be the proverbial 800-pound gorilla and threaten independent application innovation. But virtual consoles could short-circuit Microsoft's dominance and could even provide

Continued on page 109

Continued from page 106

complementary support for FireWire (IEEE 1394). Standard virtual consoles implemented in hardware could make NCs much faster and could even interface with network protocols to provide virtual consoles directly to network servers.

How could we bring virtual consoles into widespread use? We need standards, presumably sponsored and expedited by professional societies. The need for virtual consoles is well known, so standardizing them should not be a difficult task. Electrical and mechanical interfacing aside, two aspects in particular would need to be standardized: the programming and user interfaces.

The programming interface would need to be adaptable to programming and command languages, and would need to support both text and graphics applications. The user interface would need to specify a standard set of controls for switching between virtual consoles and for manipulating virtual screens.

We'd need to persuade Microsoft to adopt these standards promptly. Doing so would give the hardware manufacturers incentive to implement them. But who could effectively persuade Microsoft to adopt the virtual console idea?

Perhaps if the US computing industry got behind this proposal and if DOJ could be persuaded to compromise because the present action against Microsoft might have damaging consequences, DOJ's action could be resolved by getting Microsoft to adopt a standard for virtual consoles.

If indeed Microsoft has "created a veritable arsenal of smoking guns" ("Microsoft Accused," *The Economist*, May 23, pp. 19-21), then Klein should have plenty of leverage. ❖

Neville Holmes is a senior lecturer in the School of Computing at the University of Tasmania. He will usually, though sometimes tardily, respond to e-mail sent to neville.holmes@utas.edu.au