During the 1980s, several research universities engaged in projects to develop large-scale, distributed computing environments built around Unix. The vision was of the 3M machine, delivering what was then a whopping 1 million instructions per second, 1 megabyte of memory, and a 1-million-pixel display. The resulting environments were characterized by graphical interfaces, common services (file, mail, print), remote system management, and core middleware components (though not yet known by that name). From Carnegie Mellon’s Andrew project to MIT’s Athena, Unix was what it was all about—not only because it was the hotbed of research but also because other operating systems lacked the native network support, graphics capabilities, and development flexibility (source code) available with Unix. For computing organizations, those environments were ideal: startup costs notwithstanding, the incremental cost to support additional systems was practically nothing. So whether 100 or 10,000 systems were running Andrew, at Carnegie Mellon our cost to provide and support the environment remained relatively constant. The result was a win both for service-and-support planning and also for budget forecasting.

But anything that good couldn’t last. As DOS PCs and Macintoshes became more popular (and at so much lower apparent cost based on hardware alone), they gained a foothold and eventually formed the lion’s share of campus computing environments. Although those platforms have always had compelling advantages, distributed management and remote support were never among them. As a result, the last decade has largely felt like a black hole—supporting individual desktop computers while struggling to extend infrastructures to those systems. Some vendors recognized the need for remote management, bringing products such as Microsoft’s ZAW or Novell’s Zen to market in the past several years, but generally those systems missed the mark for the way most higher education institutions need to remotely manage systems.

So now what? Being an eternal optimist, I have to believe the darkest hour has passed. I say this because there are a few discernible glimmers of hope:

- Microsoft’s Windows 2000 and upcoming XP systems bring greater potential for infrastructure integration and more power for addressing remote system-and-software installation and management. From authenticating users through Kerberos (thanks, MIT) to defining policies through Active Directory, I can almost begin to see a time when PCs will be managed the same way that Unix systems used to be managed. At Carnegie Mellon, we deployed Windows 2000 in our public computer labs a year ago. Beyond the integration work for Kerberos and Active Directory, this required building MSI (Microsoft’s remote installation package) for every application. As with Andrew Unix support, the initial effort was significant, but the subsequent management and maintenance of the systems over the past academic year was trivial. All in all, we have a good foundation for distributed support and management. Opportunities remain to build on that foundation, to provide the degree of flexibility needed to manage environments, and to encourage vendors to adopt the MSI strategy for software installation (so that they create one MSI and each institution doesn’t have to develop its own).

- Apple’s new operating system, OS X, is built around Mach, a Unix kernel (which, coincidently was developed at Carnegie Mellon around the same time as the Andrew project). The first good news is that OS X supports Kerberos authentication. Further, the power and flexibility of the Unix kernel give rise to the possibility that even though enterprise-level remote software-management tools do not exist today, they may soon. The question is whether those capabilities will be developed on individual campuses or through collaborative efforts. Depending on the style of approach and which valuable insights might be gleaned from the Microsoft/MSI strategy, there could be a useful system down the road for Macintosh software distribution and management on campus.

- Thanks to IBM/Transarc’s Open-Source release of AFS, colleges and universities can now envision a single, transparent file system supporting Unix, Windows, and Macintosh. So a student working in a dorm room on a personally owned laptop can easily access his or her files from virtually any computer on campus. Combining a common file system with new features that vendors such as Microsoft are bringing to market, institutions can now not only provide universal file ac-

---

**Viewpoints**

By Tracy Futhey

**Coming Full Circle?**

---

62 EDUCAUSE Review September/October 2001
cess but also enable seamless roaming of workstation profiles across campus. Seamless roaming and even file sharing may not sound like new features for a campus that supports only a single operating system, but for those institutions contending with multiple flavors of Unix, Windows, and MacOS, this is a potentially major breakthrough.

So with a few bright spots to focus on, can institutions leave it to the system software vendors to complete this vision of integrating Unix, Windows, and Macintosh systems into campus infrastructures in ways that are simple to manage and support? Unfortunately, probably not. For the same reason that early vendor solutions for remote management didn’t work, I fear that the next generation of products, though almost assuredly better, probably won’t be able to meet the distinctive needs of higher education. Moreover, vendor solutions will almost certainly be proprietary. So whereas two or three systems to manage software on different platforms are certainly better than none, they’re not nearly as good as only one would be.

This brings me to what I believe might be an even brighter ray of light on the distant horizon. Is it time to begin thinking about how the glimmers of hope mentioned above might eventually support a common (dare I say “open-source”?) system for software management across heterogeneous platforms? Just as campuses now have Kerberos for authentication, IMAP4 for e-mail, and LDAP for directories, so too might institutions move toward a common infrastructure for software installation and management. Why am I even thinking it could be productive to tilt at this windmill? Here are my top five reasons the possibility is at least worth exploring:

5. There is increasing vendor recognition that partnering with higher education institutions is valuable. This should be no surprise, considering that the products the vendors are delivering today to the mass market contain more and more of the core components that defined some of the educational environments fifteen years ago and, further, that innovation in the marketplace often comes from colleges and universities.

4. Higher education institutions can leverage their environments to deliver “living laboratories” for new systems in ways that vendors cannot hope to replicate in their labs or across their entire corporations. Using campuses as a microcosm for society is a powerful motivator for corporate partners and can provide excellent research opportunities as institutions collaboratively seek to determine how software and services can be most effectively designed and eventually used.

3. It is in the individual and collective interest of colleges and universities to collaborate in solving this problem. Doing so brings together a much richer set of experiences and perspectives and also ensures that as many institutions as possible can use the eventual solution, including those that might not have tackled a project of this magnitude on their own.

2. As a collective lobbying force, higher education wields significant influence that might be used to bring competitive products together around a common underlying service. In the same way that the successes of IMAP4, Kerberos, LDAP, SNMP, and others were borne out of campus efforts to develop standards, deploy reference implementations, and lobby with vendors, this could be an area ripe for higher education involvement and influence.

1. Most important, a common solution for remotely managing heterogeneous desktop systems would allow all higher education institutions to stop throwing resources into a black hole of supporting the basic installation and management of a growing base of systems. It could also reduce the frustration and the stop-gap system administration efforts that individual computer owners face. If campuses can solve this problem in a way that lets students spend more time learning, faculty more time teaching and researching, and administrators more time keeping colleges and universities running, then this is a worthwhile effort. And from the vendor perspective, it is much the same for businesses and the home: ideally, as people start having more rather than fewer systems, they can “just get their work done” rather than dedicating significant time to managing those systems.

So at this point, there seems to be at least some expectation that the distributed management of heterogeneous systems can get better over time. And just maybe, it can take the form of a generalized solution rather than separate solutions on each campus and for each operating system. Clearly, the magnitude of the task will be large, especially when one considers the strong linkages with software licensing and security. But as I write this, anything seems possible. Maybe that’s just because the sun is shining in Pittsburgh and the Pirates won last night...