

JON FINKE

when worlds collide, 2



THE TWO-SIDED SWORD OF TECHNOLOGY INTEGRATION

Jon is a senior systems programmer at Rensselaer Polytechnic Institute, using relational databases to automate system administration and to facilitate data flow between enterprise systems. A frequent LISA author, Jon also dabbles in construction and the building trades.

finkej@rpi.edu

THE TECHNOLOGY USED TO PROVIDE telecommunication services has been evolving over the years. This often yields reduced equipment costs, increased flexibility, enhanced functions, and other good things. However, this can also drive up the complexity of our systems, increasing—or at least changing—our maintenance and support requirements, in addition to requiring expertise outside of our current staff.

This problem became apparent at our site with our new voicemail system (which uses MS Exchange as a mail store) and, to a lesser extent, our Voice over IP (VoIP) rollout.

Let me share with you a cautionary tale about the direction that early adopters are taking.

Legacy World

Looking backward for a moment, the legacy world includes equipment such as the PBX, which is based on a special-purpose computer and lots of custom hardware. This computer and its operating system had only one design function: to provide telephone service. Likewise, the voicemail system (Octel) was a dedicated, custom computer with only the single function of providing voicemail services. While these systems may have been based on general-purpose operating systems of the time, functions were much simpler then.

This approach gave the vendors a very clean environment to support and administrators a focused system to configure. They could control what operating system and application updates were needed, and, best of all, the rate of change was very low. With the exception of presumably rare bug fixes and new features, there wasn't much need for frequent updates, and those updates were driven by the application, not the operating system. In 12 years of operation, we had only a single patch for the voicemail system.

The New World

In the new paradigm (or at least the current paradigm), things have become much more complex. Our single-box voicemail system (Octel) was replaced by a collection of five machines (the Unity Voicemail server, the Exchange server, two domain controllers, and a backup server). To complicate things further, these systems are not running custom application-

specific operating systems but, rather, a general purpose operating system (Windows 2000) supplied by another vendor. We have also moved from having just a single custom application from the primary vendor to requiring additional general-purpose applications from other vendors (including Exchange, Active Directory, SQLServer, Veritas Backup, and more).

A steady stream of updates emanates from the vendors for both the operating systems and all the other applications, including the voicemail system itself. Some of these updates are bug fixes that might or might not impact our functionality; others are security patches. Unlike our old systems, these new systems operate in a network-attached world. While we may be able to use firewalls to ensure some protection, we can't always ignore the patches and bug fixes.

Our original deployment plan assumed that our new voicemail system would use the existing, supported Exchange service. Instead, we opted to install a stand-alone Windows domain and Exchange server. This was installed by the consultant who was assisting with the overall voicemail deployment, who left once things were up and running.

Now before you wonder how we bought such a troublesome product, I'd like to point out that we did all the proper reviews and evaluations; this all seemed reasonable going into deployment. A number of other voicemail systems we evaluated also provided unified messaging. We knew our old voicemail system was close to death, but then a 10-day outage due to a hardware failure (the 10 days was in part spent searching on eBay and elsewhere for replacement parts!) pushed us into a crash deployment project. The sudden shift from evaluation to installation led to some pushback from the department providing the Exchange email server, resulting in having to go it alone.

Challenges

The biggest challenge we face is that, as a department, we do not have any staff members with significant experience administering Windows 2000. Additionally, we face the same problem with some of the other things we need such as Exchange, Active Directory, and SQLServer. These are not simple, easy-to-pick-up

systems to administer. It takes a lot of time and training for someone to become proficient in maintaining these systems. In addition to applying patches and upgrades as needed, these systems need to be monitored for problems and also tuned and adjusted to keep them operating well. This seems to be a component of the total cost of ownership that wasn't quite factored in properly in our initial thoughts.

We are currently faced with error messages about memory fragmentation on our Exchange server. Certain patches for Exchange seem to address this problem. Those, in turn, may require patches and updates to the Windows OS upon which we are running the Exchange server. There have been no updates to either system since it was originally deployed. We are also faced with the problem that attempting to get support from the vendor of the voicemail system on this problem yields as a first response, "We don't support Exchange." What is more, they won't be able to provide support for their application unless all of the other components are current with updates and patches.

In researching the memory fragmentation issue, an experienced Exchange administrator found 35 technical articles from Microsoft on this issue. However, he is not available to work on our server, and no one in our department has the background and training to readily understand these articles; they are intended for someone familiar with Exchange and Active Directory.

Conclusion

Efforts are underway to address these support issues. In the meantime, we are running a production service with unpatched, unmonitored, and unsupported software.

But what's the big picture? Is every new high-tech product going to demand its own administrator (expert) or set of experts? Will these high-feature products require the large sets of components that it appears they might? I fear at this point that we're discovering a number of hidden costs for support that our experience and background did not prepare us for. Perhaps we are paying far too high a cost for the perception of better features, but the problems remain.