Early administrative and enterprise applications grew up in mainframe environments, where integrated management and monitoring tools provided status information as well as performance tracking and tuning. Answers to questions such as “How come I can’t get to the Human Resources System?” or “Why is the Student System slow?” could be ascertained by looking at the mainframe, its processes, or the network between the user and the central computer. But in today’s distributed systems, these kinds of applications depend on multiple NT and UNIX file servers, back-end database servers, front-end and back-end Web servers, distributed output devices, and sometimes application servers or transaction-processing monitors, along with all the network path dependencies among these elements. Answering the same type of questions today is a much more difficult task because of the complexity of the systems and the variety of management tools available.

Each platform in a system (NT, UNIX, Windows, network hubs and routers, etc.) might have a different tool for monitoring its operation and performance. Above the operating system, each of the services and applications might itself have a separate reporting and monitoring tool. Even though the groups responsible for the operation of each of these systems, services, and applications might be well-versed in the use of their particular tool, the information gathered may not be understood or even accessible outside of the group. For managers, tracking down the source of problems involves repeated calls and queries to each of the systems in an attempt to pinpoint the problem, with no easy way to consolidate and aggregate the information from the underlying layers.

Enterprise Systems Management (ESM) vendors attempt to solve this dilemma with a coordinated set of monitoring tools that work all the way up the protocol stack, from network connectivity and operating systems to complex enterprise resource planning (ERP) suites, and that provide a common repository for operation and performance monitoring at all levels. At the highest layer, ESM tools attempt to model and monitor business systems and practices so that the overall health of a system or application (e.g., Human Resources, Student Records, Finance) can be determined. When problems occur, drill-down tools permit recursive querying of the supporting components to identify underlying causes.

Enterprise Systems Management at Princeton University

As part of a broad implementation of new administrative systems, Princeton University began to deploy Tivoli Systems’ ESM framework and tools in 1998. Few colleges or universities had experience with these tools, largely because of the high cost, which large businesses could justify in terms of revenue growth potential and income liability, and because of the overall complexity of these systems. Most institutions have monitoring in place for their earlier mainframe solutions and have developed or acquired point-solution tools for managing networks and UNIX systems. Until the growth of new, complex systems, most have not felt the need for broader ESM tools. With Tivoli, Princeton began a partnership that provided an affordable path to determine the viability of these products in the campus environment. In addition to the systems-monitoring tools, Princeton also acquired network monitoring (NetView) as well as helpdesk (Tivoli Service Desk), job scheduling (Maestro), and output management tools (Destiny).

Our initial efforts focused on designing and implementing the underlying framework for Tivoli and establishing independent implementation of the other products. The last three were especially time-critical, since they were replacing systems being phased out, and efforts at integration took a backseat. For the initial deployment, three staff positions were loaned to the rollout effort. A year later a reorganization created a three-person ESM group to run systems management, job scheduling, and output management. In many of the large-industry Tivoli implementations, each of these efforts would have three to five people assigned to it, so our implementation took longer than...
anticipated. In addition, the rollout was slowed by staff turnover as we moved from pilot to production, by the pressure to deploy Maestro for production control, and by the need to choose an alternative vendor, Dazel, to remediate Y2K production printing (after an unsuccessful attempt at implementing Destiny). Nevertheless, we are now monitoring 149 hosts with Tivoli’s distributed management tools and framework and are using the Tivoli Event Console (TEC) and underlying database to produce regular reports and drill-down Web pages for event tracking. Production faxing and administrative output needs are being managed on 70 queues for 20 printers through Dazel, and all of the new university administrative applications are scheduled and managed through almost 300 schedules and nearly 2,000 Maestro jobs.

**Lessons Learned**

ESM software products are complex and require a large staff investment and lead time. Small organizations may have difficulty justifying the overhead and expense. And even though upper management may be convinced of the need for these products, selling the systems to line staff can be difficult. Many system administrators have their own set of “point solution” monitoring tools, which they are reluctant to abandon. Successful implementation of an ESM thus may be held up in an underlying catch-22: system administrators see little value added, and yet the value comes only when all the individual systems can be monitored and an overall aggregated view presented. A hybrid approach—wrapping the point solutions to incorporate them into the ESM framework—can leverage existing monitoring and get buy-in once the benefits of aggregation can be demonstrated.

Systems like Tivoli provide rich functionality and robustness at the cost of complexity; a distributed object structure allows Tivoli agents to function independently and at a layer of abstraction that allows for complex aggregation and analysis. The TEC, which monitors log activity, provides a rich correlation engine to provide higher-level analysis. And although graphical user interface tools and rule-builders simplify much of the interaction with these components, working with the underlying object structure, creating programmatic command-line interactions with the system, and digging into the Prolog code of the TEC’s rules engine are almost essential to a successful implementation.

**Staffing**

Expect to dedicate staff to your ESM efforts: the systems are complex and training is essential. Outside consultants can speed up architecture and installation issues, since few of these systems operate “out of the box,” and an experienced engineer can help steer you around many potential pitfalls.

A large percentage of ESM efforts fail, in large part because of the complexity of these systems. Establishing targeted six-month goals and concentrating efforts on particular phases of the roll-out or systems to monitor can increase chances of success. These phases can include overall systems architecture, framework deployment, UNIX distributed monitoring, job scheduling, output management, NT distributed monitoring, database monitoring, Web and e-mail monitoring, and finally, administrative systems monitoring.

Is it worth it? Several years into the ESM deployment at Princeton, we have begun to notice anomalies and problems that our local monitors didn’t track. And we’re starting to tackle our databases and higher-level systems. The up-front effort to implement the system was high relative to the benefits of existing monitoring, and much momentum was lost in time needed to implement the necessary but peripheral job scheduling and output management systems. But we are about to roll out two major distributed applications (Human Resources and Student Systems) in addition to the current Finance, Accounts Receivables, and Alumni Systems. And project managers are now beginning to ask for the type of monitoring and management that can be obtained only within an ESM framework.

**How to Get Started**

You can find demonstration run-throughs on the Web sites of the largest ESM vendors: Tivoli (http://www.tivoli.com), Computer Associates (http://www.computerassociates.com), and BMC (http://www.bmc.com). Closer to home, creating an inventory of existing “point solution” monitors on your systems is a valuable starting point. There are likely far more of these than you know, and these are the areas that need to be converted over or “wrapped” if ESM is to be successful. You may have difficulty getting system administrators to give up their existing tools, but the big win comes when all the monitoring efforts are consolidated into a central repository—where they can be more easily shared, used by higher-level management, and combined to provide analyses of complicated systems and applications.

Lastly, organizations considering ESM should be sure that they understand what goals they want to achieve. Getting buy-in and early cooperation across the organization will help steer the many decision processes along the way and increase chances for successful implementation.

Daniel J. Oberst is Director, CIT Enterprise Services, at Princeton University.