# Managing the Mission Critical Environment

E. Scott Mentor
Enterprise Systems Management Corporation

# Managing the Mission Critical Environment

*E. Scott Menter* – Enterprise Systems Management Corporation

## ABSTRACT

Mission critical environments present an entirely unique set of challenges to the systems manager. ‘‘Make it work and watch the budget’’ are the standing orders for most systems administrators, and yet in some situations the consequences of systems-related problems can be devastating, whereas in others the result is no more than inconvenience. As UNIX systems are increasingly at the heart of vital financial, medical, and defense technologies, we thought it would be useful to examine how systems administrators can cope with the challenges of managing such an environment. Finding and deploying the correct technology is only half the story: the systems administrator in this role is part of a social dynamic in which personnel and their skills are constantly being evaluated against their ability to support the mission. Thus, to be successful, the systems administrator has to be equally adept at handling both the politics and the technology that are endemic to the mission critical environment.

### Defining the Mission Critical Environment

Every professional systems administrator works in a mission critical environment, to the extent that her job depends on the secure and reliable operation of systems and networks. Indeed, we expect that every systems manager reading this paper will recognize some elements of their daily lives, and therein (we hope!) lies the general appeal of our message. Still, something in particular is meant by the term ‘‘mission critical’’ as used by marketing organizations and want ads, so we’ll try to come up with a clear definition and stick with it.

A mission critical environment is one in which even a brief interruption of service can have a serious effect on the ability of an organization to operate. We’ve used vague terms like ‘‘brief’’ and ‘‘serious’’ on purpose: there is a wide and fuzzy border between ‘‘critical’’ and ‘‘non-critical’’ environments. Still, like so much in life, you know a mission critical system or system component when you see it. We’re reminded of the tiny transistor that recently failed in a NASA Mars probe, leading to the total loss of that billion dollar system. Another example might be a computer monitoring system located within a hospital’s intensive care unit.

UNIX systems, once thought to be too quirky and unreliable for mission critical environments, can today be found in places that take security and reliability very seriously indeed. Among the earliest and perhaps most illustrative of these is the Wall Street trading floor, the site of billions of dollars of transactions each day. The ability of the Sun, HP, IBM, and other workstations to handle this load directly affects the function of the financial markets in which they operate, and therefore also the quality of life of pretty much everybody. Don’t make the mistake of thinking that if the machines stopped working the old manual methods could be hurriedly put back into place: even if there were anybody around to remember those procedures, they could never cope with the volume of traffic associated with today’s markets.

If you were to wander into a mission critical systems environment based on UNIX workstations and servers, what could you expect to find? We’ve noted some characteristics that are common to many of these sites:

- Beepers – In a truly mission critical situation, the staff (especially the systems administrators) may be called upon to perform some feat of technical heroism at any hour of the day or night.
- Attitude – Users have one. Managers have one. Developers have one. And systems administrators have one too. The business relies on the technology, and everybody’s job is on the line. We’ve noted a severe sense-of-humor deficiency in many mission critical environments.
- Backups – Buying a piece of hardware? Better buy two, in case the first one fails. Oh, and what will we do if the network goes down? What about building power? What if sun spots trash our satellite connection to Tokyo? ‘‘No single point of failure’’ is the rallying cry of mission critical systems designers. Sometimes they add: ‘‘regardless of expense.’’
- Vendors – Lots of ’em. And they’re hungry. They hear ‘‘regardless of expense’’ and come running. For systems administrators, this stampede has proved particularly frustrating as vendors in our market have been pushing

solutions that don't seem quite ready for prime time.

- Scale – Though not a necessary component of mission critical environments, many of these sites are deploying and managing systems on a large scale. And, most businesses will balk at simply hiring more and more systems administrators to handle the volume. The question in mission critical sites is how to manage large numbers of systems, often in geographically remote locations, without creating a huge demand for technical staff.

Into this situation wanders the happy (or perhaps hapless) systems administrator, who must learn to cope in a high-visibility, high-exposure situation while keeping his life – and his career – intact.

### The Systems Administrator in the Mission Critical Environment

Our company is comprised of systems administrators (SAs) who spend their professional lives managing mission critical UNIX environments. We've noticed that just about all of our activities can be categorized under two headings: technology and politics.

It's easy to understand the technology part: after all, we're UNIX systems administrators, and we're constantly writing shell scripts, evaluating new hardware and software, inventing mechanisms for configuration management, etc. All of us got into this business because we were attracted to the technology, and we are making our living from our understanding of how that technology is best deployed and managed.

The need to be so involved with politics is less obvious to the non-SA. It can be readily explained, though. Systems administrators are in the business of managing expensive, important, and sometimes scarce resources. In many companies the allocation of those resources, like the allocation of profit and loss, is directly related to the political fortunes of the various parts of the business. How many times have we been approached with this sort of instruction: ''You know those workstations that just arrived that George Jetson ordered for the engineering department? Well, we're going to 'loan' them to Barney Rubble at the sales department.'' The rise and fall of managers and organizations are often expressed in terms of who gets the new machines and who is assigned the more experienced technical support staff.

There is also a more subtle political aspect to the job of the systems administrator. Too often, the SA is faced with dueling priorities: conflicting tasks requested by end users. In a typical situation, one user will ask the SA to perform some function immediately, such as restoring a file from backups. Another user will ask the same SA to perform some other task immediately, such as clearing up a performance problem with a file server. In these situations, the systems administrator is suddenly called upon to make decisions that should be based on corporate priorities (about which he may be largely in the dark), but will most likely be based on who screamed the loudest, or which problem seems more interesting, or some other arbitrary basis.

Our intention in preparing this paper is to examine the special challenges of systems management in the mission critical environment. By viewing the situation from the dual perspectives of technology and politics, we hope to establish a reference point that will be useful to systems administrators currently working in this type of environment, and those who are considering moving to it. In order to do that, the rest of the paper will review some of the common functions associated with systems management – design, technology selection, deployment, operations, policies and procedures, and managing systems management – in the context of these two categories. The material represents an overview of observations drawn from our experience with various mission critical systems environments. Where possible, we'll present hints for preserving personal and professional happiness while working in the mission critical fast lane.

### Design

Systems administrators are often involved in the design phase of infrastructure, applications, and certainly, systems management architectures. In terms of technology, there is real pressure in the mission critical environment to create redundant systems, systems that continue to work even in the event of a component failure. And yet, redundancy can often lead to overly complex designs that are actually *more* likely to fail due to their complexity, in spite of whatever failure modes were devised. Additionally, when legacy systems are included in the design, complexity is very difficult to avoid. In these situations, the use of a ''gateway'' system that isolates the points of contact between the new and old architectures can be very useful in keeping the design as simple as possible. In one of our current customer accounts we are helping to design redundant gateway systems between the workstations and the mainframes: the gateways themselves are usually vital components of a mission critical system.

Political pressures are certainly present in the design phase. Often, there is a fight to be involved in the design phase of a large project, or a desire to get ''buy-in'' from virtually every part of the enterprise. The systems administrators can find themselves left out of certain designs in which they ought to have early and extensive involvement; for example, new office or datacenter facilities, locally developed applications, or network architectures. Once the design is in progress, another important political question arises: is somebody being designed out of a job? Systems administrators have to ensure their presence in important design efforts, and show sensitivity to the question of whose livelihood may be affected by their plans.

### Technology Selection

Typically, systems administrators will play a key role in the evaluation and acquisition of technology for the enterprise. In mission critical environments, the question of where to aim on the technology curve can be very important. On Wall Street, for example, individual departments were moving to abandon the security and comfort of the mainframe for the young and untested UNIX workstation environment in the mid to late 1980s, while MIS organizations in the same firms viewed this new technology as a threat. (They were right: many of those who fought hardest against the introduction of UNIX in the 80s find themselves in different jobs today.) Thus, the evaluation and acquisition stage can be a big mess of conflicting technical and political requirements.

From a purely technical standpoint, the important considerations today during this phase are *openness* and *interoperability*. Because definitions for these terms vary widely from one organization to the next, and because virtually all vendors will tell you that their products are bountifully endowed with both features, hitting the mark can be difficult for the technology selector. Systems administrators are usually in the best position to evaluate what will work best with what, as they tend to have a broader view of how software, hardware, and infrastructure interact than do other staff members.

The political dimension of technology selection can often be seen in the struggle over ''build vs. buy'' decisions. Particularly in mission critical environments, one often encounters a distrust of off-the-shelf software. If you're betting the business, as the reasoning goes, you want to be able to trust your applications, and (here's the flaw) the only applications you can really trust are those you build yourself. In truth, we feel that many companies have squandered millions on home grown solutions that could have as easily been implemented using software from the local Egghead outlet. In any event, systems administrators will be called on to make either solution fit into the environment, and will want a say in what vendors are reliable and what management features should be included in locally developed software.

Choosing the right vendor is as important as choosing the right product, of course. Carol Kubicki makes this point in her paper, [Kubicki92], ''...[vendors] that might come into contact with your customers on your behalf should be held to the same standards as your own organization...[When the vendor fails to perform] the customer becomes more dissatisfied with the [systems] administration group who is ultimately responsible.'' The author is referring to field service personnel, but the same is true for any vendor, particularly in the mission critical environment.

### Deployment

For some reason, the technology available for systems deployment seems to be behind that for other areas of systems administration. We assume that the reason for this is that each type of system has its own peculiar way of being installed: even various models of workstations from the same vendor can require different installation techniques. In the mission critical environment there is often a requirement to begin an installation after the users go home on Friday night and have everything reliably up and running before users return on Monday morning. As a result, the installation mechanisms provided by vendors are frequently insufficient to the task. In order to do the job well and in the brief time allotted, we have had to develop automated installation mechanisms that run *unattended*. The requirements for and design of such a system is worthy of its own paper: in this one we'll just point out that an unattended installation mechanism requires that all configuration information (names, IP addresses, etc.) be available in advance of the actual installation. The obvious reason is that if the machine is being installed without a person nearby, the answers to configuration questions have to be obtainable elsewhere. It turns out that recording this data in advance is a good idea anyway: installation time is the wrong time to be selecting names and configuring name service domains.

In the political arena, deployment is often the first contact between systems administrators and the ultimate end users of the machines and networks under their care. Because it's important that this first encounter be positive for all involved, it's vital that systems administrators be able to recognize a fully functional workstation when they see one. That statement is not as trite as it first appears. At installation time, there is often a great deal of successive approximation of workstation functionality. Here's what happens: the workstation is installed, the SA tests it, it seems to work (she can log in), and everything seems to be ready for the user. The user comes in in the morning and finds he can't log in. He calls the systems administrator, who discovers that the user isn't yet included in the NIS domain for that workstation, so she fixes that problem. A while later, the user calls back: he can log in, and most things seem to work, but he can't run Lotus. Within minutes, the SA has taken care of the situation, and is congratulating herself on responding so quickly to the user's problems. The phone rings: Lotus works, but when I try to print, nothing happens...

## Operations

In contrast to the deployment phase, the operations phase has had considerable attention from vendors in terms of available software solutions. Lately a great deal of noise has been generated about the ability of systems management and network management tools to be *integrated*. This integration includes help desk tools, performance monitoring applications, problem tracking systems, and cable management tools, as well as the traditional event monitoring features normally associated with network management. While full integration of all these tools may be a desirable goal, the systems administrator has to be careful not to sacrifice functionality on the altar of systems integration. Especially in the mission critical environment, it's more essential that, for example, the network management tool be able to detect problems and generate alerts than that it automatically create trouble tickets in the problem tracking system. It's not that automated trouble ticket generation isn't a desirable thing; rather, the problem is that trying to get too many things working at once can delay the delivery of the whole system.

The operation of mission critical systems can be a high-intensity job. By definition (ours), these systems can't go down even briefly without adversely affecting the ability of the enterprise to stay in business. Political challenges aside (we'll get to that shortly), just making the technology work right can be a real effort. Therefore, the goals for managing configurations in this environment have to include *consistency*, *simplicity*, *predictability*, *auditability*, and *reversibility*.

- Consistency.
  In the mission critical environment, it's more important than ever that the same task is performed the same way every time. It's the rare site today that can claim that, for example, adding a user happens the same way no matter what systems administrator is on duty or what else is going on at the same moment. When you're busy it's surely tempting to toss a line into the `/etc/passwd` file, do a quick `mkdir`; `chown` and be done with it, figuring you can include the niceties later.
  Of course, ''adding a user'' may be too broad an operation to be done the same way every time. There may be derivative operations, like ''adding a user to the systems administration group,'' or ''adding a user to the sales domain'' that qualify as tasks subject to the consistency requirement. The granularity can be adjusted to the environment, subject to the requirement of simplicity.
- Simplicity
  The importance of simplicity has been noted earlier in this paper; in this context, we mean that there should be a minimum number of configurations in the domain of managed objects. It's still hard to accomplish this long-understood goal. Workstations, for example, still have a small number of files that must be unique from machine to machine. Other types of configurations are frequently overlooked. At one site I've worked at there has been a great emphasis in minimizing router configurations, which generally have all sorts of uniqueness built into them.
- Predictability
  A non-deterministic system is clearly unsuited to a mission critical role. If, to revisit our earlier example, I invoke the procedure to add a new user, can I predict *exactly* what's going to happen? When we spoke of consistency, we talked about things happening the same way each time: predictability is about knowing *what* it is that's happening, and what the effects will be on the overall system.
- Auditability
  Well, we know what's happened, we know it happened the same way that it always does, but sometimes we also need to know *who* did it and *when* they did it. Also, we want to know what derivative operations or options were involved. In mission critical environments this ability is very important: identifying what happened and when can be crucial when the enterprise has been hit with a systems failure.
- Reversibility
  Sure, you know how to add a new user. What about removing that user? Do you know where all the files owned by that user are – even the ones that don't live in his home directory? What about mailing lists? Automount points? Any task that is performed has to be reversible, or the system will soon become unmanageable.

From a political standpoint, it is during the operations phase that things usually get hot. In a mission critical environment, everybody is quite serious about performance, reliability, and security. The systems administrator has to deal not only with the technical challenges involved in meeting those requirements, but also with the perception of the end users. During the famous Internet worm incident, the head guy at the site I was working at was extremely agitated about the possibility of ''infection,'' in spite of the fact that we were not at the time network-connected. Knowing what your users are thinking is a very important part of getting operations right: that point is the major thrust of [Kubicki92].

It is also at this stage of things that users become very concerned with the organization of the systems management staff. We are big proponents of a modified centralized systems management model. Actually, our view is not so different from that of

Peg Schafer [Schafer93], despite that author's emphasis on the decentralized nature of her model. In fact, the model proposed in that paper is essentially a centralized one. The precise details of the structure we promote, and the differences between our view and that of [Schafer93] are left for another paper. The important thing to note in this context is that systems administrators in a mission critical environment have to be extremely sensitive to the immediacy of user requirements, and that those who are not may find themselves under relentless fire from the business. If a centralized group fails to be responsive to user needs, the result is not a nicely balanced structure like that presented in [Schafer93], but rather a complete dismemberment of the organization leading to total decentralization of the systems management function. This in turn can lead to increasing problems with security, standards, and efficiency for the business, and ultimately end in the failure of UNIX technology in an enterprise.

## Policies and Procedures

Most systems administrators have the responsibility not only for maintaining hardware and software but also for helping to design the policies and procedures that form the day to day guidelines for running a large site. The important thing about policies and procedures in any site is that they be entirely consistent with that site's mission and objectives. In the mission critical environment, policies and procedures have to be developed that support and reinforce the vital nature of the operation: systems administrators have to be very flexible about how things work, because the mission may require quick adjustments to new situations.

From a technology standpoint, we've noticed something important: procedures that aren't encoded into software are not useful. Forget documenting your procedures: it's a waste of time. No two systems administrators will follow the same written procedures the same way every time. Procedures that are encoded into software can be powerful tools for keeping a large environment under control, while actually parceling out various bits of responsibility to non systems administrators. For example, if one could add a new user to their domain by simply typing

```
add_user lastname firstname machine
```

then it would no longer require a systems administrator to add a new user. Furthermore, the software could record who ran it, how often, and exactly what happened, for later perusal. And, systems administrators could confidently predict the outcome of each occurrence, because the function of the software is well known and trusted. In short, all the conditions we described earlier for reliable operation of a mission critical environment require procedures implemented into software in order to be fully met.

There is another important point to make about the politics of policies and procedures. As we've pointed out, they have to be flexible to accommodate changing business requirements. The important thing to remember, though, is that to the extent that ethical considerations are involved there should be no flexibility. There should still be a few things more important to the individual systems administrator than the mission of her organization. We mention this because it is all too common for businesses with mission critical systems to ask their systems administrators to take unethical actions in the name of defending the enterprise, and too often, the systems administrators involved don't even recognize that there is a moral issue involved. More education will help here, and we have high hopes that SAGE can further discussion on this problem.

## Managing Systems Management

Much of today's technology is advertised as making it easier to manage departments and even entire enterprises. Middle management positions are falling by the wayside as technology enables widespread communication within and across departments, co-opting many middle management functions. Yet, one of the key supporters of this trend, the systems administration manager, is sometimes least adept at gleaning key information from his own systems.

An important feature of much of the world's technology is its ability to *informate*, a term coined in [Zuboff88]. The expression refers to the ability of systems to provide information as a side effect of their ability to *automate*. For example, when we develop and install custom software for our clients, we generally include a counter that tracks how many times the application is executed. As a result, we can demonstrate the value of the software to our clients by showing them a monotonically increasing usage curve. Similarly, many of the management tools used today by systems administrators can be exploited to provide valuable information. Want to justify the expense of that sendmail tutorial to the boss? Click, click, you've produced a report showing a steady increase in the number of email-related service requests submitted. Systems administration managers, like all other managers, have to use the best tools available to analyze their department's performance and to justify requests for resources, increased budgets, etc.

From a political standpoint, the recommendation of [Kubicki92] that customer surveys can be an important part of the management process makes a lot of sense. The issue of organization is felt most keenly by the systems administration manager, of course: demonstrating good service through system-generated information, survey results, and met commitments will go a long way towards supporting whatever structure the SA manager wants to put into place. Finally, and particularly problematic in mission critical environments, is the issue of junior staff. It is entirely common for junior staff to be thrown directly into a critical support area to sink or swim. In fact, just this approach was recently promoted by [Nather93]. It's the responsibility of the systems administration manager to make sure that junior staff receive the direction and support they require in order to succeed: nobody wins when one of these young SAs burns out or becomes disillusioned with his career. Messages of the ''how can I get out of being a systems administrator'' variety are all too common on Usenet.

## Final Words of Wisdom

Although we hear a lot from academic sites about the work they're doing in this or that area of systems management, it turns out that some of the most useful and interesting stuff is being created and used in mission critical environments. The motivation to provide good solutions is simply higher at such sites. Of course, one also doesn't get exposed much to the results of these efforts, because mission critical environments are usually proprietary environments, and the business keeps its secrets to itself.

For systems administrators contemplating entering the world of mission critical systems, we have a bit of advice. Make sure a high-pressure, hard working environment appeals to you, and keep in mind that any chinks in your technical or political armor may be examined publicly and at length by those you are seeking to support. And finally, make absolutely sure *before* you accept any job that the enterprise at which you're interviewing has practices and policies that conform to your own ethical code.

## Author Information

Scott Menter gave up the hectic but remunerative life of a Wall Street technical manager for the hectic but poorly compensated life of a southern California entrepreneur. His company, Enterprise Systems Management Corporation, provides technical and management expertise to companies with really large or widely distributed UNIX workstation installations. After getting his computer science degree from Brandeis University in 1985, he worked at various commercial and academic sites as a systems administrator, ending up in charge of the worldwide systems and network management department for Lehman Brothers. Finding few investment banks to work for in Orange County, Scott founded Enterprise Systems Management after he left New York in 1992. Today he splits his time between running the company and convincing Usenet readers that systems administration really is an okay way to make a living. Reach him electronically at escott@esm.com or telephonically at +1 714/573-4075.

## Bibliography

[Kubicki92] Kubicki, C. *Customer Satisfaction Metrics and Measurement*, LISA Conference Proceedings, 1992.

[Schafer93] Schafer, P. ''A Proposed New Model of Large Site System Administration'', *;login: The USENIX Association Newsletter*, Vol 18, No. 1, Jan/Feb 1993, p. 18.

[Zuboff88] Zuboff, S. *In the Age of the Smart Machine*, Basic Books, Inc., 1988.

[Nather93] Nather, W. ''Think or Thwim: The Cold Creek Approach to Systems Administration Training'', *;login: The USENIX Association Newsletter*, Vol. 18, No. 4, Jul/Aug 1993, p. 22.