Managing Open Access Labs: “MacGyver Principle”

Mark A. Adams
University Computing and
Telecommunications
University of Houston – Clear Lake
Houston, TX 77058
(281) 283-2946
adamsmark@cl.uh.edu

ABSTRACT

For a variety of reasons, in recent years doing more with less has become standard operating practice within IT departments of higher education. Colleges and Universities, following the industry trend to decentralize aspects of enterprise management to smaller functional areas within the organization, are moving less centralized enterprise management functions to individual departments. The use of Microsoft's Active Directory structure with Organizational Units (OU) and the other object concepts taken from Object-Oriented Programming philosophies has solidified this push.

Keeping individual functional areas current with technology trends can be difficult when the central IT organization is still working in a legacy environment. These areas are relegated to finding ways to offer new services without the benefit of the new technologies that make the services possible. For an area like the Open Access Lab environment of a university, the staff is often forced to adopt the "MacGyver Principle:" making something useful or functional out of inconspicuous and otherwise overlooked elements.

While many in IT would call this ability troubleshooting, the reference here conveys more than just troubleshooting: it illustrates the challenges placed on an Open Access Lab Staff when the technologies and services modeled after recent market trends are not available yet the faculty, staff and student body request these technologies and services.

Categories and Subject Descriptors

K.6.4 [**Management of Computing and Information Systems**]: System Management - *Centralization/decentralization*.

H.1.0 [**Models and Principles**]:

General – *“MacGyver Principle”*

SIGUCCS’04, October 10-13, 2004, Baltimore, Maryland, USA.
Copyright 2004 ACM 1-58113-869-5/04/0010…$5.00.

General Terms

Management, Performance, Reliability, Standardization, Security, Theory

Keywords

MacGyver Principle, MacGyver Toolkit, Troubleshooting, “Universal Troubleshooting Process”, methodology, “The Six Thinking Hats”, OS compatibility, automation and enterprise management, system lockdown procedures, Ghost Console, Deep Freeze, migration, innovation, Open Access Labs, Open Access Computing, roaming user profiles (RUPS), Interactive Development Environment (IDE), Microsoft Management Console (MMC), Organizational Unit (OU), Group Policy Object (GPO)

1. INTRODUCTION

Without belaboring the pros and cons of decentralization in IT, especially in an academic environment, it is important to understand how the decisions made by the IT administration can influence the individual department’s ability to provide services. In fact, the results of these decisions are no more important to the main clients; the faculty member using the technology to teach and the students using the technology to fulfill requirements and to facilitate their learning. As the faculty continue to update lesson plans to stay current with technology and the students look for more efficient ways to meet their academic requirements the need for IT departments to remain dynamic and proactive becomes more vital each day.

The problem arises for the individual IT functional department like Open Access Computing when the demands from faculty and the student body conflict with the limited resources and services the department can provide given budgetary constraints and IT administrative directives. In order to alleviate the effects of these convergent concerns, the organization must learn to “think outside the box”. A functional department like Open Access Computing must go beyond troubleshooting. The real goal is to design a new paradigm that will help keep pace with technology changes. One technique we have named to help us is the MacGyver Principle. This paper focuses on our experiences in the Open Access Computing environment and how we have incorporated this new philosophy into our procedures.

2. MACGYVER PRINCIPLE VS. TROUBLESHOOTING

As IT professionals we all deal with troubleshooting. For the lucky ones, troubleshooting may be limited to either software or hardware. Unfortunately, the IT professionals working in the Open Access Computing environment must troubleshoot both software and hardware. Additionally, these professionals' troubleshooting demands may even require them to go beyond hardware issues related to just the workstation, i.e. all imaginable peripherals and networking equipment. In many respects, the professionals responsible for lab environments are more dynamic than the IT professionals in specialized fields such as programming and web development. This substantiates both the importance of these professionals to the goals of an academic institution and the wise use of these professionals as conduits of 180 degree communication between the hourly computer technician and salaried professional in the IT structure of the university.

Because of the extent of troubleshooting required of the personnel in the Open Access Computing environment, it is essential that these professionals have a concrete troubleshooting methodology to follow. As the content of the paper is related to the idea of the MacGyver Principle and our examples of its use, I defer ideas and well established troubleshooting methodologies to others who are more qualified than I. In my on-going novice efforts to become a professional troubleshooter, I have found many expert resources on the Internet. One website which I have found to be a plethora of information is Steve Litt's Troubleshooters.com. At this site you will find links to his online magazine, Troubleshooting Professional Magazine and his troubleshooting methodology, "Universal Troubleshooting Process" (UTP) [1]. Additionally, I have found Morris Rosenthal's website of computer repair flowcharts to be of great help in solidifying our troubleshooting methodology [2].

So the question arises, "How does a troubleshooting methodology, used to find the best solution to a problem, differ from the MacGyver Principle?" Strictly from a philosophical standpoint, I believe the difference rests with the notion of innovation. While a practiced troubleshooting methodology will secure a viable solution to a recognized problem and may turn up some interesting possible solutions, sometimes the eventual outcome of the troubleshooting methodology may lead the organization to believe the only viable solution is to upgrade a legacy system through expenditures.

In an organization where the budget does not permit and the troubleshooting methodology has suggested expenditures, what recourse remains? It falls back to the organization to find a way. The old adage, "Where there is a will there is a way", rings true where the troubleshooting methodology fails. This is where the MacGyver Principle steps in, making something useful or functional out of inconspicuous and otherwise overlooked elements. The principle is simply a suggestion or recognition of the fact that innovative thinking is required to solve this problem. You could argue that the MacGyver Principle can be nestled into Litt's UTP or any other troubleshooting methodology. The goal here is simply to pay due respect to the import of thinking outside the box. We have found the incorporation of this extension to our troubleshooting methodology invaluable.

As with the varied examples on the Internet about troubleshooting methodologies there are equally as many methodologies surrounding the facilitation of innovative thinking processes such as Dr. Edward de Bono's ideas on innovative thinking, "The Six Thinking Hats" [3]. Although our idea, The MacGyver Principle, is a rudimentary expression of innovation with the name borrowed from a familiar cultural reference for the intrinsic entertainment value, the main message still carries weight; "thinking outside the box" can foster creative solutions to major problems.

3. DEMONSTRATING THE MACGYVER PRINCIPLE

The following are some limited examples of our work-arounds or innovations to materialize out of our use of the MacGyver Principle:

3.1 Software and OS Compatibility Concerns

The most unique aspect of working in an Open Access Computing environment is the absolute need to build a standard and reliable computer image for all users. In order to achieve this primary goal of image building a comprehensive cloning methodology becomes paramount to the overall success of an Open Access Computing environment. No matter how perfect you think your process is, there is always room for improvement. Needless to say, there is no perfect process! Your current imaging and cloning processes would be perfect only if the workstations did not require software updates, software installations, configuration changes, or OS migrations. If all these factors remained consistent, there would be no need for an Open Access Computing Staff. Since this will never be the case, you will have to modify your imaging and cloning processes every semester because you will inevitably get a new software installation request from a faculty member or a scheduled software update in the form of a version change or software fix.

Because of the dynamic nature of image building, to a greater extent the whole technology industry, you will suffer from some sort of software incompatibility issue on your lab image. Sometimes a configuration change or permission change is all that is required to fix the incompatibility. On the other hand, what do you do if the above options do not work and the organization does not have the money to purchase the required number of licenses for each system so that you can migrate to the newest and greatest software package to fix your incompatibility problems? Or, worse yet, the software package you recently purchased created an unforeseen incompatibility issue on your system? We experienced the later scenario just recently.

As often happens in the Open Access Computing environment, a school within the university makes a software purchase without coordinating with the lab staff personnel. Often the faculty representative making the purchase is not aware of the licensing issues or the complexities surrounding roaming user profiles

(RUPS) with non administrative privileges, and therefore often makes the wrong purchase. Apart from purchasing choices, it may be that all the correct decisions were made and the software still generates unforeseen incompatibilities on your system. This was our case.

After talking to the vendor of a recently purchased software package we assigned full permissions to Users in the registry key HKEY_CLASSES_ROOT to resolve the user access errors we were receiving. The solution did resolve that error; however, it created an incompatibility in yet another software vendor product. The fix for this was to change our imaging methodology yet again to account for a change in our software installation and configuration sequence. This example may play more to a good troubleshooting paradigm than the MacGyver Principle, as no new use of otherwise disparate tools were purposed. However, the problem epitomizes the complexity of software incompatibility issues.

Another and more distant example of a system incompatibility issue surrounds the inherent problem of Microsoft's Interactive Development Environments, like the former Visual Studio 6, which often require full local administrator privileges to utilize all of the functionalities of the product. In an academic environment where faculty members try to facilitate and encourage the adoption of new tools in an effort to incorporate the most recent market trends, the Open Access Computing environment has to balance these requests with concerns of network security and system lockdown procedures. With the Visual Studio 6 example we took matters into our own hands and created an executable program to modify the API components toolset in Visual Basic 6 so that all users, not just the local administrator of the workstation, could call these functions into Visual Basic 6 when they needed them. This solution was outside our normal duties but we had troubleshooters who were able to program a solution. In this sense our team was innovative in its approach, implementing the MacGyver Principle. Later, when the budget permitted, we resolved the problem by migrating from the Visual Studio 6 IDE environment to the new Microsoft Visual Studio.Net IDE environment.

Without doubt, software compatibility issues will arise when your image requires, for the most part, all the software packages of each school in your institution. Apart from the size of your image you will always have to deal with the added complexity of dealing with the requirement that each regular user have access to all the necessary functions of each program although many software vendors write the programs to only function under the local administrator account. Keep in mind that each of these opportunities will allow you to capitalize on the MacGyver Principle.

3.2 Roaming Profile Concerns

As indicated above, RUPS always adds to the complexity of the Open Access Computing environment. Apart from the software compatibility issues presented above, another problem with RUPS are the reasonable constraints put on the profile size limits. It is not hard to imagine the storage space required on the institution's file servers to accommodate each user in the organization. A problem arises when users fill up their profiles. Some programs assume a great deal of the user's profile and cause random errors. Some users find that their data and files are not synchronized to their profile on the file server. Often the user can not logoff the system, which requires the user to perform a warm reboot, loosing any work not backed up on removable media.

We encountered one software package which hogs the user's profile space, Macromedia Studio MX. In our efforts to troubleshoot this problem we were unsuccessful in finding a solution through typical means, i.e. configuration changes, permission changes or modifications to the installation sequence of our image building methodology. In efforts to explore security and lockdown procedures we had stumbled onto a crude script for folder redirection. With a few modifications to the script referencing the appropriate Macromedia Studio MX files that were filling up the user profiles we were able to redirect some of the files to the local system with the creation of a VB program and incorporate it into the Ghost Console as a task for deployment to all lab systems therefore reducing the amount of space that Macromedia Studio MX assumed of the user's roaming profile, allowing the users to save their work and then logoff properly.

This innovative use of both programming skills and system automation tools like Ghost in combination to resolve a problem epitomizes the idea behind the MacGyver Principle. Currently, in an effort to minimize and motivate users to be less dependent on their profiles for storing essential data and files, we are promoting USB flash drive technology as a form of file and data backup. We have gone to great efforts to purchase new workstations with easy access to multiple USB ports and writeable CD drives.

3.3 Network Security and PC Lockdown Concerns

As Microsoft's time to counter virus and worm attacks promulgated by black hats on the Internet shrinks to the zero-hour, network administration staffs in the IT industry justifiably say no to any request from clients that might open up a port to possible compromise. The only problem with this tactic is it conveys the message that the cracker/terrorist has already won.

When your network administration staff locks down the network so tight that critical services are being denied then the idea of a network becomes obsolete. There has to be a balance. I maintain that the balance comes with education. Many services can be secured; it is only a matter of how to secure them. The ready answer of "no" to a practical service request should not be the norm. Still, understanding these major challenges to network security, I find some services do solicit a fair amount of skepticism and caution before deployment or activation.

The most recent challenge to our university's IT staff came with the deployment and use of Microsoft's Internet Information Services 6.0 (IIS 6.0), key to Microsoft's .NET Framework initiative. From the perspective of programming faculty members who like to design lesson plans that capitalize on Visual Studio .NET's ASP capabilities, the client IIS 6.0 component is required at minimum. The security risks of previous IIS versions are well known to most network administration staffs. However, IIS 6.0 has gone a long way to create a more secure environment. Microsoft has even created an IIS Lockdown tool and the Microsoft Baseline Security Analyzer (MBSA) which can be used to find areas susceptible to crackers and to aid in the configuration of your systems and IIS Web Servers against unauthorized use. These tools can be found at Microsoft's Security Guidance Center web site. Admittedly, we have not used these tools, having only become aware of them recently, and we do not offer this information as a statement of support, only a resource. With respect to our environment we were only looking to implement IIS services on the client workstations so that the student could create and test ASP.NET web applications and services on the local system and intranet.

Due to the fear of the default anonymous user account, port accessibility to crackers and past experiences with the IIS predecessors our network administration opted to deny this service, thereby restricting the faculty member's instruction in the classroom. From an Open Access Computing standpoint we were required to configure all of our workstations without the IIS component. This presented a problem with the installation of Visual Studio.NET, which preferred that the IIS component be installed prior to its installation. It is possible to bypass the recommended installation of IIS and proceed with Visual Studio.NET installation, however, the ASP functionality will be lost and the process to add the IIS component afterward becomes more difficult and uncertain with respect to the master image.

In our anticipation of an evidential decision to allow limited IIS functionality (local system and intranet only) we decided to install the IIS component and disable the service in the Services MMC on the client systems. This solution allowed for a smooth installation of Visual Studio.NET while adhering to the network administration's denial policy on IIS. Interestingly enough, after several months of faculty requests, the service was allowed. Not only was IIS enabled on the client systems, a departmental IIS Web Server was also setup to allow for intranet services with all the necessary authentication and port security measures in place.

As a first line, customer service, functional department, the Open Access Computing Staff implemented the MacGyver Principle with respect to recognizing and validating the needs of the faculty and looking outside the box to facilitate an organizational change in philosophy by pushing the necessity of IIS to the mission of the Management Information Systems and Computer Science programs at the university and finding the services work-around until a paradigm change occurred.

A second area of concern for our network administration was the accessibility of Terminal Services like Remote Desktop across campus. Again concern surrounding the likelihood of a port being compromised was of major concern. As lab staff we were inclined to use Remote Desktop to perform some maintenance and minor configuration changes on the fly to our client systems. Again the policy was to deny this capability. As a result, a practical solution for our environment was to utilize the open source, freeware product RealVNC. This provided the same service as the Remote Desktop and allowed us control over the systems we deployed the product on as local administrators. As with the IIS issue, Remote Desktop was allowed once the network administration staff felt reasonably sure that the network security was in place and that ports were locked down from unauthorized requests. With both IIS and Remote Desktop presenting new concerns and challenges, the Open Access Computing Staff had to look for alternatives that presented viable solutions to the understandable restrictions and polices placed on the staff from the network administration department.

3.4 Automation and PC Enterprise Management Concerns

One idea can summarize all the previous examples from the standpoint of our Open Access Computing environment, enterprise automation. In our efforts to define and create methodologies that facilitate our management of the Open Access Labs and all our computers, we have pursued products and services that allow us to have greater control over all aspects of computer maintenance and management. With respect to our imaging and cloning methodologies we chose to utilize Symantec's product Enterprise Ghost 7.5. This product has proven to be invaluable. Another product we are hoping to use in assisting us with better management of our systems is Faronics's Deep Freeze software. This product will assist us in resolving issues surrounding limited privileges assigned to the average user, who sometimes needs administrative control over certain application packages.

In an environment with the need to create one common user experience across multiple systems with roaming user profiles and a myriad of dynamic application configurations the challenge has always been, how do we allow a user all the functionality of an application without compromising the system or the network? This question really remains the most important question for all of us in the Open Access Computing environment. What makes it difficult to find solutions that meet these goals is that it becomes a fine balancing act between meeting the customers' needs and securing systems and networks. Add to this complexity the fact that your organization may be using legacy systems or may still be on an NT network. Without the new advantages of Organizational Units (OUs) and Group Policy Objects (GPOs) that a current network environment like Active Directory provides, or domain administrator privileges or tools like Systems Management Server (SMS) or the new Software Update Services (SUS), the goal of enterprise automation and management becomes very difficult for an Open Access Computing Staff.

Alternatives like Ghost Console and Deep Freeze virtual partitions help to counter the shortcomings of a legacy network environment, allowing the Open Access Computing Staff some enterprise control of their lab environment and systems. Whether you rely on Litt's UTP to lead you to finding solutions to problems or not; rest assured, you will find yourself relying on the MacGyver Principle at some point as a tool to promote the balance between the customers' needs and the network's need for security.

4. MACGYVER TOOLKIT

As a teenager I can recall many times when my father would ask me to fetch his toolkit from the garage so he could fix something my brother and I had managed to break in one of many battles unleashed upon our home. This familiar context of the word toolkit has become antiquated with the Information Technology Era. Now the term refers not only to actual objects or tools but also to the abstract concept of one's mental abilities. As an IT professional I am never wanting for a how-to or certification book from Barnes and Noble or the local computer super-store. My main problem is trying to determine which of the million and one publications will address my specific question or offset a lacking skill. And yes, the same problem exists when surfing the Internet…thank goodness for Google.

As you might guess I deal with many vendors and of course each vendor has a web presence. Sometimes I find the answer for one vendor's problem product on an unrelated vendor's website, usually trying to get that vendor's product to play nice with another vendor's product while living in the Microsoft home. Of course, I will go to Mr. Microsoft or his representative, Microsoft TechNet, if the vendors don't play nice. Usually Mr. Microsoft only thinks of himself.

Apart from the previous tools, the MacGyver Toolkit should include many abstract tools related to the troubleshooting methodology and the MacGyver-like methodology of your institution. Also don't forget those wonderful vendor compact disks. They come with useful files and utilities designed to aid in troubleshooting some of the more common issues. Needless to say, if each teammate of the troubleshooting team does not work together to solve the issues and find practical and creative solutions utilizing their own MacGyver Toolkit then the only solutions will be mediocre at best and more than likely costly.

5. CONCLUSION

Whether your academic institution's philosophy espouses decentralizing or centralizing your IT administration responsibilities, one thing remains certain. The success of your institution will revolve around its people. For the IT departments of your university the ability of your IT professionals to incorporate creative thinking into their troubleshooting paradigms, building the ultimate MacGyver Toolkit will be essential in insuring they remain dynamic professionals ready to meet the challenges of an ever changing environment of technology.

In order to facilitate troubleshooting and the incorporation of the MacGyver Principle within an Open Access Computing environment we offer the following summarized table indicating the areas of concern, the problems identified, the viable solutions, and the different conventions of the MacGyver Principle utilized.

Different conventions of the MacGyver Principle

Concern	**Problem**	**Solution**	**Convention**
OS and software to software compatibility	Vendor to vendor software conflict	Redefine imaging methodology	*Breaking out of your paradigms*
RUPS and profile space constraints	Vendor product fills up the RUP	Program folder redirection and create a Ghost Console task	*Combining disparate tools*
Security and IIS 6.0	IT administrative denial of service directive for an essential component	Work around directive until IT administrative directive is modified through education	*Influencing other's paradigms*
Security and Enterprise Automation	IT administrative directive denying Remote Desktop and restricting enterprise automation	Look for free alternative technology that provides the same functions	*Adopting alternative technologies or solutions*

It is worth saying, "Necessity is the mother of invention." And therefore, I believe the MacGyver Principle could, if given time and the correct motivation find a way to render the explosive (referencing the trite ticking time-bomb or, in modern terms, a dirty bomb) inert by clapping and producing sound ways…or something like that. The point being, there is usually a reasonable work-around. Allow your troubleshooting methodology to get you there but don't forget that the MacGyver Principle can fill in where your troubleshooting process leaves off.

6. ACKNOWLEDGMENTS

I would like to thank the following co-workers in Academic Computing for their editing and suggestions for this paper: Raed Abilmouna, Diane Yee, and Sana Zeidan.

7. REFERENCES

[1] Litt, Steve. *The Universal Troubleshooting Process.* Troubleshooters.com, 1996-2004.

[2] Rosenthal, Morris. *Computer Repair with Diagnostic Flowcharts.* Foner Books, 2004, Fonerbooks.com.

[3] Dr. de Bono, Edward. *The Six Thinking Hats.* Edwdebono.com.