ETD MINI-GRANT Final Report

Proof of Concept: A Novel On-line Learning Approach for Electrical and Computer Engineering Technology

PROJECT DIRECTOR: Mohammed Abdallah
Assistant Professor
Phone: 315-792-7544
Fax: (315)792-7399
Email: abdallm@sunyit.edu

INSTITUTION: State University of New York Institute of Technology SUNYIT
Engineering Technology Department
100 Seymour Road, Utica, NY 13502 -1311

Date: 02/01/2014
Abstract

Advanced information technology provides students access to a wide range of learning experiences. One of the challenges of the on-line forum is exposure to hands-on experience. There currently exist disparate components of technology that are suited to address specific needs in the Engineering, Math, & Science (ESM) as well as the Engineering Technology (ET) departments. The crux of this proposal is to investigate and validate, at the educational unit level, a new concept of bring the state-of-the-art technology to enhance pedagogical on-line learning approaches for both students and faculty which will enable us to make a recommendation for adoption to a college level, university wide, and possibly beyond.

Creating a dynamic and efficient learning environment with ubiquitous access to computing and laboratory resources is the main goal in this proposal. Additionally, this new alternative environment is expected to provide enhancements to the current on-line instructional model in electrical and computer engineering technology programs. Furthermore, it will be an effective test bed for new educational technology of core interest to SUNYIT and ETD. Finally, it encourages faculty, staff and administrators to capitalize on campus expertise to enhance the ETD by developing and sharing innovative teaching and learning methods.

The proposed idea is separating hardware from software. The software will be in the campus/server-side. A school will host it including the license and upgrading issues. The hardware will be managed via students. Students will be able to locally connect their hardware to their computers and still be able to remotely access the developing tools.

Justification

In electrical and computer engineering technology programs, hands-on experience is very significant. To gain this experience, a developing tool (such as Microcontroller CodeWarrior IDE or FPGA Quartus IDE) and an attached hardware (such as a Dragon 12 Microcontroller or Altera Cyclone II FPGA evaluation boards) are needed. Most of the developing tools are expensive for students and need to be purchased, installed, and hosted by schools.
On-line learning is the optimal solution for many students. However, current on-line learning approaches lack the hands-on experience. Hardware is always at the campus/server-side. To gain hands-on experience, a student should be on-campus to access the available hardware in labs. Hardware means any equipment that students need to connect to certain software such as microprocessor boards, data acquisition device or even a medical device. To date, there are three solutions to provide some kind of hands-on experience in on-line learning. First, simulation can be used. Students can rely on simulation tools installed on their computers or they can remotely access simulation tools which are hosted at their schools. In both cases, students do not gain the hands-on experience that is available to on-campus students. Second, some schools invite their students few times on-campus to expose them to the hardware and provide the necessary hands-on labs. It is not a practical solution since not all students will be able to make it. Third, students can remotely access the developing tools which are hosted at schools but the problem is the attached hardware will not be accessed. Students are not able to touch the connected hardware, give inputs, or even see and monitor the outputs as shown in Figure (1-a).

Therefore, the principal director proposes a novel on-line learning approach as shown in Figure (1-b). The idea is separating hardware from software. The software will be in the campus/server-side. A school will handle it including the license and upgrading issues. The hardware will be handled via students. Students will be able to locally connect their hardware to their computers and still be able to remotely access the developing tools. The challenge is to create a module that can have an image of any installed developing tool at a school side and provide this image to any authorized students who remotely accesses the school server. The ultimate advantage of the proposed approach is students can remotely access the expensive developing tools hosted by their school and locally connect the required hardware to gain the necessary hands-on experience. Moreover, it enables instructors to change a course to include a state-of-the-art hardware without changing any infrastructure on campus.
Project Phases

Initially, the PI had three potential solutions to prove that the proposed idea can be done. The first potential solution is the Thin Client. The second potential solution is the Windows Terminal Services. The third potential solution is the VMWare environment. The PI investigated each and every one of these solutions. The challenges were difficult. The PI could not find any IT member who thinks that it can be done. The main challenge was how the data and information will be sent via USB port and the remote computer should think that the hardware is local.

In the next sections, the three potential solutions will be discussed in more details. The PI found the first two solutions do not work. Finally, the third solution based on the VMWare environment works perfectly. The concept has been proven.

1. THIN CLIENT

- A thin client, sometimes called a lean client, is a low-cost, centrally-managed computer devoid of CD-ROM players, diskette drives, and expansion slots.

- The term derives from the fact that small computers in networks tend to be clients and not servers. Since the idea is to limit the capabilities of these computers to only essential applications, they tend to be purchased and remain "thin" in terms of the client applications they include.

- A thin client is a network computer without a hard disk drive. They act as a simple terminal to the server and require constant communication with the server as well.

- Thin clients provide a desktop experience in environments where the end user has a well-defined and regular number of tasks for which the system is used.

- As software as a service (SaaS) gains popularity, it is expected that thin clients and blade PCs will replace desktop PCs in many work and educational environments. In general,
they are not as vulnerable to malware attacks, have a longer life cycle, use less power and are less expensive to purchase.

✔ A thin client only sends keystrokes, mouse clicks and images to the server. This makes for more secure storage and makes thin clients more secure than desktop computers.

✔ Thin client is also used to describe software applications that use the client-server model where the server performs all the processing.

**THIN CLIENT QUICK ANALYSIS**

✔ Easy to deploy as they require no extra or specialized software installation.
✔ Needs to validate with the server after data capture.
✔ If the server goes down, data collection is halted as the client needs constant communication with the server.
✔ Clients run only and exactly as specified by the server.
✔ Portability in that all applications are on the server so any workstation can access.
✔ Opportunity to use older, outdated PCs as clients.
✔ Reduced security threat.
✔ Thin clients are less expensive, seldom need to be upgraded, and are less prone to breakdowns.

**HOW TO SET UP A THIN CLIENT ENVIRONMENT**

✔ Decide on an operating system for your server. Most thin client environments use Microsoft or Linux on the server.

✔ Purchase a server that has the necessary memory, hard drive capacity and processor speed to run your thin client workstations.

✔ Determine the software required by the people using the workstations, including email, word processing programs, spreadsheets and other typical applications. Then ensure you have the updated software and required software licenses.

✔ Create a test network by connecting the server to one or more thin clients and following the steps below. Only after you are satisfied that the server is working as required should you attempt to install the server and workstations into a working environment.

**Linux Environment**

✔ Install the Linux operating system onto the server.

✔ Install terminal server software onto the server.

✔ Install other required software programs. These could include Open Office, a word processor, and spreadsheet, presentation software suite equivalent to Microsoft Office. SAMBA is a software that enables file sharing with Microsoft servers. R desktop is a program that allows your network to connect with Microsoft Windows Terminal Server. Evolution is an email program used with Linux.
Microsoft Environment

✓ Install Windows Server 2008 onto the server, including Windows Terminal Server.
✓ Install Microsoft Platform Builder onto the server. This comes with most thin client workstations.
✓ Launch Platform Builder from the start menu and follow the installation guide.
✓ Install any necessary applications for your thin client workstations. These typically include applications like Microsoft Word, Excel and Outlook.

Network Configuration

✓ Select a firewall to protect the server from hackers and malicious software on the Internet
✓ Select a switch to allow you to connect each thin client and the server together.
✓ Connect the firewall to the Internet, and connect the switch to the firewall. Connect the server and thin client workstations to the switch using 10/100 Ethernet cables.

TYPES OF THIN CLIENT

Linux Thin Client

✓ A thin client computing environment based on a Linux operating system is known as a Linux thin client. A Linux thin client environment offers the benefits of often being free to use and also being open source for the operating system to be customized.
✓ There are various flavors of Linux thin client environments, including thin station. One recent advantage to running a Linux thin client environment is the possibility of running another operating system, such as Windows, inside of a light Linux installation. This can be done using the free Virtual Box software.

Windows TS Thin Client

✓ Microsoft offers Terminal Services, a thin client server-side solution for Windows computing environments. Terminal Services is available on Microsoft Windows Server operating systems. Using the service on a network enables multiple client computers to access and use shared software applications, such as Microsoft Office and run them locally on a client computer.
✓ Windows TS is based on Win Frame, a product from Citrix, which was a repackaging of Windows NT. Microsoft licensed the technology and made it part of future versions of the NT operating system.
✓ This is also the basis for the Remote Desktop Protocol used in Windows operating systems today, which allows for graphical thin client access to Windows computers.

VMware Thin Client

✓ VMware has been spearheading much of the modern thin client technologies.
Their software VMware View is the company's thin client application for servers and desktop clients. With VMware, multiple types of desktop operating systems (including Windows and Linux) can be stored on a single server and accessed by multiple client computers.

VMware is unique in focusing on support for multiple client operating systems, rather than its own proprietary operating system. This means it is operating system independent and this can be beneficial for organizations running multiple operating systems.

**THIN CLIENT SERVER INSTALLATION REQUIREMENTS**

**MEMORY REQUIREMENTS**

- In a Windows environment, Microsoft says that the thin-client system software and the client together require at least 6 MB of RAM.
- If the thin client is booted from ROM, it requires a minimum of 4 MB of ROM.

**KEYBOARD REQUIREMENTS**

- The keyboard subsystem should support all the functionality that the operating environment requires. For instance, it should support some important function keys (F1 to F12).

**Pointing Device Requirements**

- There are some requirements for a mouse or touchscreen that you use with the thin-client server. The device should imitate mouse movement and button presses. The device should also support a minimum amount of picture-image quality and sound.

**ADVANTAGES OF THIN CLIENT**

- Configuration of the clients is very basic and many manufacturers provide a means of zero touch configuration.
- Terminal servers do not require much more attention than a 'normal' PC meaning that only one computer must be looked after instead of 30 (the approximate number of clients that a server will support).
- The centralized location of the terminal servers also makes maintenance considerably easier because there is no need to travel around site to perform common tasks.
- An added benefit of thin client computing is that the infrastructure provides a ready-made remote access solution, giving access to whatever applications you have, after hours.

**SECURITY**

- Security is improved in a variety of important ways.
✓ Data is stored on the server, not the client device. This means that CIOs can worry less about client computers, hard drives or thumb drives being stolen or otherwise compromised.
✓ In addition, CIOs can focus their attention on securing the servers that power thin clients. It is much easier and less costly to secure a manageable number of servers in a data center than to protect data stored on thousands of widely distributed devices. This architecture also makes it simpler to ensure continuity of operations.

**COST FACTOR**
✓ Thin devices cost roughly half as much as a PC and require replacement roughly half as often, making upgrades much less expensive.
✓ Thin devices have fewer components that can malfunction, servicing them is less costly.
✓ In a thin client architecture, most software resides on the servers, not the clients, significantly reducing the labor involved for upgrades and patches. CIOs realize the full benefits of remote client management from the data center.
✓ To summarize, thin clients have a lower total cost of ownership than PCs.

**POWER SAVER**
✓ Thin client infrastructure is also more environmentally friendly because it uses fewer components and can operate on less power.
✓ With today’s focus on green IT, this becomes both a cost and political advantage over more traditional PC usage.

**Lower cost of hardware**
✓ With thin-client you can reuse all your existing workstations if they are still in running order. This saves you about $1,000 to $2,000 plus GST per workstation including installation. Alternatively you may replace them with dumb terminals costing between $400 to about $720 (with a 17” LCD monitor). There is almost no installation cost involved.
✓ If you have old workstations that will need to be replaced sooner rather than later, you can simply recycle them because a terminal server will allow you to run these old workstations at the same speed as a brand new workstation.
✓ Because thin-client terminals (or recycled workstations) do not have to have speed, it also means that you can hang on to them far longer. A typical PC workstation lasts between 3 to 4 years. A dumb terminal can last 5 to 6 years at least.

**Lower cost of support**
✓ With standard workstations, if you need to make any changes to the software (eg upgrade MS Office) you need to go around each workstation
one by one and update them on-site. With thin-clients you change it once on the server.

✓ As a result of all this, your total cost of ownership is a lot lower with thin-clients than traditional fat-client PCs.

**Lower cost of workstation failure**

✓ In the event of a PC failure in a fat-client environment, if the machine has to be replaced or rebuilt, there is a lot of downtime and costs involved. It can take up to two days to rebuild an old PC or configure a new one. This is because with fat-clients all the old files and programs have to be transferred from the old machine to the new and the new machine then has to be customized to the user.

✓ Thin client, one strategy is to keep a spare dumb terminal ($400) or old machine in storage. In the event of a terminal failure, all you have to do is to replace the faulty one with the spare terminal (by unplugging the old one and plugging the new one in, an operation that should take no more than five minutes) and the user is up and running again. The new terminal will retain all the programs, data, customizations, even the last document being worked on before the old terminal failed, without any intervention by any technical staff.

✓ This results in savings in user downtime (takes five minutes instead of up to two days) and technical service costs (the replacement can be done by just about anybody instead of a qualified engineer).

**DISADVANTAGES OF THIN CLIENT TECHNIQUE**

**The Server issues**

✓ One of the main disadvantages of a thin client computer is the server itself. Since thin clients do not process anything or store any data on their own, they require a connection to a server to handle these tasks for them.

✓ Unlike a standard desktop computer, or thick client, should the server go down or suffer from hardware failure, all connected thin clients are affected, and they will not work.

✓ Desktops, on the other hand, can still work in a local computing capacity without network resources should connected servers fail.

✓ There are also some problems with certain software titles that have a tendency to grab as many resources as possible. This normally isn't a problem on a single user system but when you are sharing the resources with 30 or so other users, everybody will experience a performance hit.

**Network Bandwidth Issues**

✓ Since thin clients require a connection to a server, they are also dependent on the networking infrastructure. Should the network suffer from bottlenecks, lags or outages, the thin clients work at a minimal capacity, if they can function at all.
✓ As with issues associated with the server going down, if thin clients lose their connection to the server due to a network outage, they will not work.

**Lack of Multimedia Support**

✓ Media-rich applications like audio and video require a great deal of network resources as well as computing power to play.

✓ With thin clients, computing power is shared among all the clients connected to the computer. Bandwidth used to deliver packets between the server and thin client is also shared between all connections.

✓ Due to these resources being at a premium in a thin-client setting, resource-rich multimedia applications do not run effectively on these machines.

**Software licensing requirements**

✓ Some software licenses are not allowed to be used on a terminal server. For example, Microsoft Office OEM licenses cannot be run on a terminal server. Only OLP (Open Licenses) or FPP (Full Priced Package) MS Office licenses are permitted in such situations. If you are presently running MS Office OEM software and wish to migrate to a thin-client network, please discuss your options with your IT provider.

**Performance degradation:**

✓ The performance of thin clients greatly depends on how powerful the servers are. If you use low quality hardware or software, the performance of the client is likely to be degraded. Thin computing offers the productivity required by users at a lower cost than fat clients without compromising manageability or security. They are environmental friendly since they emit low heat and carbon, and use low power, which accounts for their prolonged life span.

**EXAMPLES OF THIN CLIENT SOURCE**

**Native Windows Terminal Services**

Native Terminal Services is an excellent way to begin looking at thin client technologies and is perfect for small scale deployments. Server 2008 has also added several of the features that were only previously found in Citrix products such as the ability to publish applications rather than just desktops.

**What you need**

✓ Terminal Services CAL for each device of user
✓ Windows Server CAL for each device or user
✓ Windows Server license for each terminal server
✓ A decent spec server for roughly every 30 concurrent connections

**Advantages/Disadvantages (assuming Server 2003)**

✓ Cost effective way of implementing thin clients
✓ Clients available for most common operating systems (although many clients have been reverse engineered by open source projects)
- Limited load balancing support
- No concept of a farm (group of servers) which means you have to specify which server to connect to

**Windows and Citrix**

Citrix XenApp (formerly Presentation Server) gives the most features and possibilities for managing your implementation by far. Of course, it is not the perfect fit for all implementations - particularly small scale ones - and is one of the more expensive solutions.

**What you need (REQUIREMENTS)**

- Terminal services CAL for each device or user
- Citrix CAL for each concurrent connection
- Windows Server license for each terminal server
- A decent spec server for roughly every 30 concurrent connections (pref. an additional server for a little redundancy)

**Advantages/Disadvantages (assuming Enterprise or greater licenses)**

- Load balancing across servers based on any number of metrics (e.g. number of connections, CPU load, memory utilization, etc.)
- Clever installation management
- Client software for most OSs (e.g. Windows, Mac, Linux, Unix, Pocket PC, Java Web client - all supported by Citrix)
- Secure Gateway allows publishing of the farm to the Internet through a reverse proxy
- Too many management and monitoring tools to mention
- Licensing is relatively expensive (upwards of £120 per concurrent license)

---

**2. WINDOWS TERMINAL SERVICES**

Terminal Services has been a part of the Windows server OS since NT 4.0, but its history as a technology goes back further to the late 80s when Ed Iacobucci founded Citrix. Ed had been working with IBM trying to develop a multi-user version of OS/2, but IBM wasn't much interested so Ed left to start his own company (good move). In the early 90s Microsoft came out with Windows NT 3.51, their first really solid 32-bit Windows server platform, but Microsoft too wasn't much interested in developing a multi-user version of their OS at the time, so in 1994 Microsoft granted Citrix access to their NT source code so Citrix could develop this technology as a third-party vendor. Soon afterward Citrix released their revolutionary Win Frame product, and its immediate success soon led Microsoft to shift gears and in 1997 Microsoft licensed Citrix technology so they could incorporate it into NT 4.0. The result was the release in 1998 of Hydra, Microsoft's code name for Windows NT 4.0 Terminal Server Edition. Hydra of course has since been superceded by the built-in Terminal Services feature of Windows 2000 Server and more recently in Windows Server 2003.

**The Terminal Services Advantage**
From the operations side of running a business, there are really only two things that are important: time and money. The great thing about Terminal Services is that it can save you both.

From a time-saving perspective, Terminal Services lets administrators install, configure, manage and maintain applications centrally on a few servers. This is usually much faster and easier to do than deploying applications on hundreds or thousands of desktop machines at different sites across an enterprise. And by making the job of the administrator easier, companies also save money on IT support costs for their information systems infrastructure. Furthermore, centrally-deployed applications are usually easier to maintain (for example, patching and upgrading) and simpler to troubleshoot when things go wrong. As a result, downtime is reduced, users are more productive, and business booms along.

Another cost-saving perspective is that since in a terminal server environment all application logic runs on the server, the processing and storage requirements for client machines are minimal. This means you can save money by keeping in service older desktop computers running legacy versions of Windows, and focus your limited IT budget on a few high-powered systems to run as your terminal servers. Or you can toss your old desktop PCs and buy thin clients like Windows-Based Terminals (WBTs) instead, such as those from Neoware, Wyse, and other vendors. The options are almost limitless since terminal servers let you run almost any DOS, Win32, or Web-based application from almost any client platform.

There are two major players in the landscape, Microsoft and Citrix, and we'll start by looking at Microsoft's latest offering, the built-in Terminal Services component of Windows Server 2003.

**Windows Terminal Services**

First off, let's take note that Terminal Services in Windows Server 2003 is light-years ahead of its predecessor in Windows 2000 Server. Key enhancements over that older platform include the following:

- Remote administration mode no longer requires the installation of additional components, you just select a checkbox on the Remote tab of the System tool in Control Panel and presto, up to two users can simultaneously connect to your server to remotely administer it. Not only that, you can even connect to the actual console session on the remote server, something you couldn't do in Windows 2000.

- Remote Desktop Protocol (RDP) has been enhanced to improve display and device redirection and enhance security through powerful encryption algorithms, making Terminal Services so secure you don't need a VPN tunnel when you're connecting to it over a public network like the Internet.

- The optional Session Directory component now lets you scale Terminal Services upwards to meet the demands of even large enterprises by letting you build a load-balanced terminal server farm that lets users reconnect to the same terminal server they were connected to should they somehow become disconnected. This maintaining of session state lets you run mission-critical business applications on terminal server farms.

- And perhaps best of all, Terminal Services can now be configured, managed and locked-down using dozens of Group Policy settings new to Windows Server 2003, allowing administrators to
take advantage of the flexibility and power of Group Policy to simplify the configuration and management of Windows Terminal servers.

So it would seem that if you want to deploy terminal servers in the enterprise that Windows Server 2003 is a no-brainer as far as choice is concerned, right? After all, it's got the horsepower, it's got the scalability, and it’s even got the clients for different Windows platforms ranging from Windows 95 to Windows XP to Windows CE on Pocket PC devices. There is even a client for the Macintosh platform, and also an ActiveX client that runs within the Internet Explorer web browser so you can connect to Terminal Services over the Internet. For more details on Windows Terminal Services and how to set it up and configure it.

3. VMWare Environment

This is the working solution. One has to download the VMware server on the server at school and the VMware client on the students’ computers. In the next few figures, snapshots have been taken to show the successful process of separating the hardware from the software.
## Budget

<table>
<thead>
<tr>
<th>Item</th>
<th>ETD Funding</th>
<th>Match Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servers and UPS and trial VMware licensee</td>
<td>$1,500.00</td>
<td>$2,500.00</td>
</tr>
<tr>
<td>Windows Remote Desktop Licenses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Travel to 120th ASEE Annual conference, June 22-27, 2013</td>
<td>$2,000.00</td>
<td></td>
</tr>
<tr>
<td>Compensation</td>
<td>$2,000.00</td>
<td></td>
</tr>
</tbody>
</table>