

SNAPSHOT OF A DEDICATED COMPUTER LAB

Barb Hawes

Department of Mathematics and Computer Science

Southwest Minnesota State University

Marshall, MN 56258

hawes@southwestmsu.edu

Abstract

Southwest Minnesota State University has partnered with The Schwan Food Company to create an Information Technology Learning Center for the Computer Science department. The author, who is the administrator of this dedicated computer lab, is interested in participating with faculty, students and other computer service professionals in a dialog about the best way to meet the unique computing needs of the Computer Science curriculum. With that purpose in mind, this paper will describe the current configuration of the IT Learning Center and share questions, problems and solutions that have been encountered by the lab administrator.

1 Introduction

Southwest Minnesota State University has partnered with The Schwan Food Company to create an Information Technology Learning Center for the Computer Science department. The purpose of this learning center is to meet the unique computing needs of the Computer Science curriculum while providing students various applied experiences in the field of information technology. While the mission is simply stated, it has raised many questions for the learning center administrator, the author. What are the unique computing needs of the Computer Science curriculum? What is the best way to meet those needs? How do you balance the interests of the students with the system administration needs of the learning center? Because it was decided that the lab would work with the campus network rather than be completely separated from it, all decisions about hardware and software must meet the security requirements of Computer Services and integrate with the existing networking environment. This creates a second layer of questions. How can the security and integrity of the campus network be maintained while providing the tools and resources that Computer Science faculty need to support their courses? How can the equipment for the learning center, which is all Sun Microsystems and UNIX based, be integrated with the existing computing infrastructure, which is all Intel and Windows based?

In the five years that the learning center has existed, these questions have been refined rather than answered. The roles of the people involved have taken shape. The Computer Science faculty defines what the needs are and manages the money allocated for supporting the learning center. The director of Computer Services oversees purchasing of equipment, installation of software and integration of services by drawing together staff of Computer Services as needed. Students are becoming more involved in the learning center as programming lab assistants, system testers, and researchers into new technologies. The administrator of the learning center serves as a liaison between the groups making sure that system needs and issues are clearly communicated, managed and improved. While the learning center has begun to fulfill its mission, sharing the current configuration and future plans for the learning center with other Computer Science educators and information technology professionals will enable the lab administrator to participate in the ongoing dialog about computing technology in the Computer Science curriculum.

2 Physical Resources

Currently, there are two labs devoted to supporting the Computer Science curriculum. The first lab is the SMSU-Schwans Information Technology Learning Center, here after referred to SITLC (see Figure 1). It is located in a room that was previously used as a mathematics lab. There are 18 SunRay 1 thin client terminals in the lab. Work areas exist which can be easily set up for special projects. Any servers that need to be available to student assistants are also located in this lab. These include a database server, printer

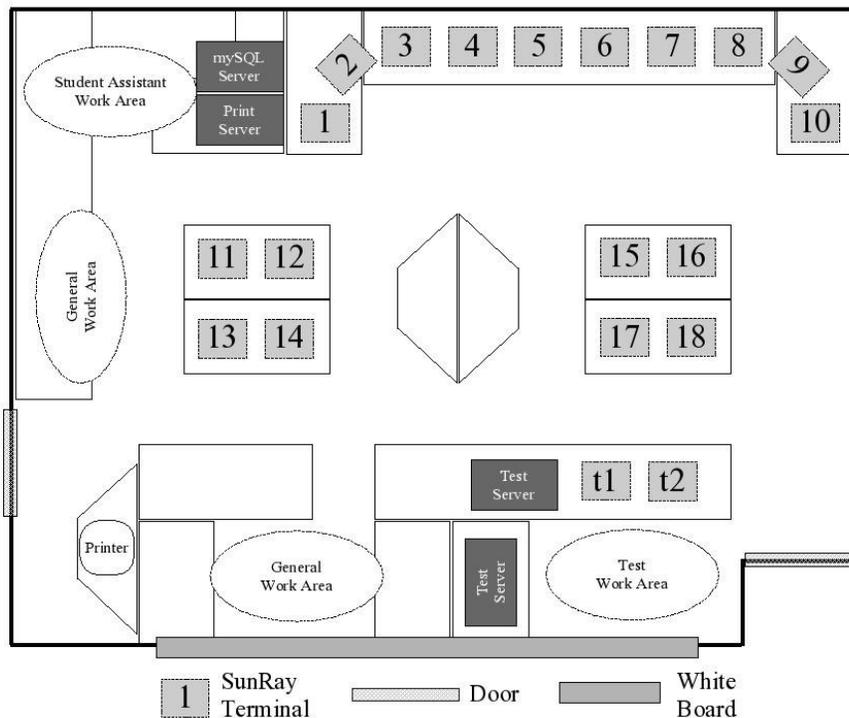


Figure 1: SITLC Layout.

server and a SunRay test environment. The lab is open from 8 a.m. to 10 p.m. during the semester. Lab assistants offer help to students at various posted times during the week.

A second lab was created in the former location of a mainframe lab. This lab, the teaching lab, is only open when faculty reserves it for their courses (see Figure 2). There are 22 SunRay 1 thin clients available for students which limits the size of the classes that use this lab. The equipment in the teaching lab is not secured because the lab is locked unless being used for a class. An ELMO, a SunRay 1 thin client, audio/visual equipment and a laptop connection, all connected to an overhead projector, are available at an instructor station. Each faculty member also has a SunRay 1 thin client in their office facilitating the preparation of demonstrations for classes using the teaching lab. Though these 5 SunRay terminals connect to the SITLC network rather than the teaching lab network, the user experiences no differences.

A single E420R server manages all 46 SunRay 1 thin clients (see Figure 3). Each lab has its own dedicated Cisco switch connected to a PCI Fast Ethernet card on the E420R server. A third on board Fast Ethernet interface connects the server to the campus network, allowing all thin clients Internet access. There are two mirrored 36 G SCSI hard drives in the server which hold the root, /var, /opt and swap file systems. This server is at its maximum configuration with four 450Mhz UltraSPARC II processors and 4G of RAM.

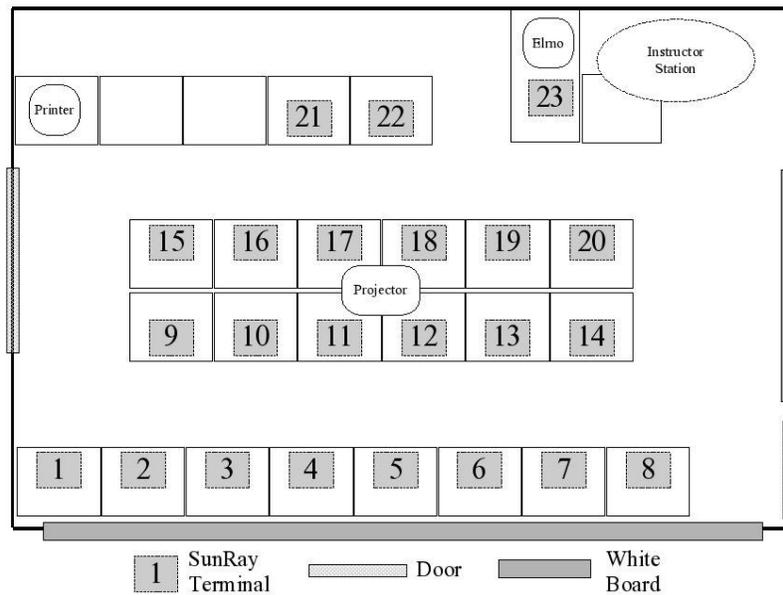


Figure 2: Teaching Lab Layout.

An A1000 StorEdge disk array with eight 18G hard drives is connected to the E420R via SCSI interface. This disk array holds home directories for faculty, students and courses. Each group has its own file system to isolate problems and simplify backups and restores. Faculty control course directories that give them combined storage space on two separate file systems of up to 68G. Because no quotas are imposed on faculty, they must be considerate of how much space they use. So far, no conflicts have come up. There is 50G of space available for student home directories. Students are given hard limits of 200M and soft limits of 500M to prevent any one student from filling up this file system. This would pose a problem if sixty percent of the students all used their maximum soft limits at the same time, but the odds of this happening seemed slim. If all students are at their hard limit, there is still additional space on the file system. This balances student need for storage space with the needs of the entire system.

As was mentioned previously, a SunRay test environment is located in the SITLC. An Ultra 10 workstation runs identical software as the E420R server. Ideally, the file systems would be set up identically. However, because there is not a duplicate disk array, some differences with file system configurations exist. Otherwise, the Ultra10 SunRay server is configured as much like the E420R as possible, including a DLT8000 tape drive. This provides an environment to test patches, software installations and configurations, and backups before performing them on the production environment. Students who will be responsible for administering support servers, like the database or print servers, can

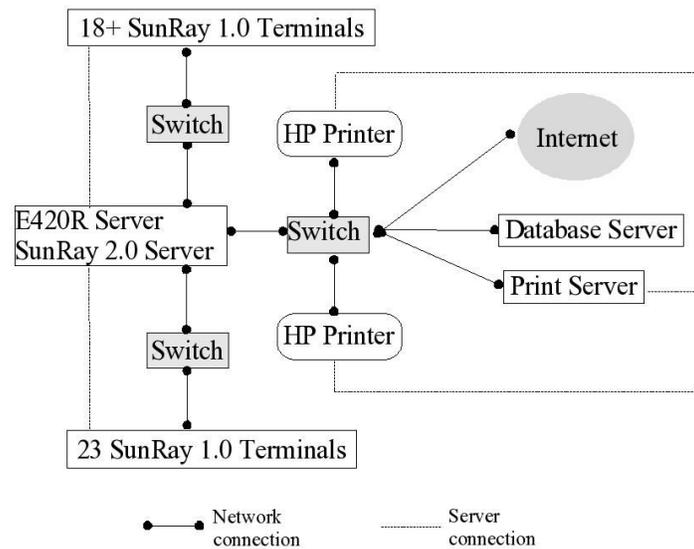


Figure 3: Network Topology.

learn system administration tasks on the test environment. Once they are proficient, they can then manage such servers in the production environment.

Other available hardware resources include a number of Sun workstations, two Sun Cobalt servers, over twenty additional SunRay 1 thin clients, and three eight port switches. Two of the SunBlade workstations are currently used for an HP print server and a mySQL database server. The switches and other workstations have been combined to mimic small networks for a variety of special projects run by students, including distributed processing, firewalls, and intrusion detection systems. Such projects are arranged at the request of students who want to explore the topic for a course or honors project. Typically, only the minimal software required by the project is installed. This makes it fairly simple to set up the environments. Students may be involved either through the system set up or only after the system is ready to go, depending on the scope of the project and their skills.

Responsibility for the labs and their equipment falls to the SITLC lab administrator. This person is accountable to both Computer Services and faculty in the Computer Science department. The administrator manages the E420R server and all attached hardware. Students do not have physical access to this server which is located in Computer Services and treated as any other campus server. Other, support servers, like the print and database servers, are physically available to the students in the SITLC. Student lab assistants are given permission to the servers to perform maintenance tasks as dictated by their system administration abilities. When no student assistants are available to support these servers, the lab administrator assumes responsibility for them. The SunRay test environment is

partially available to the students, with access being granted on a project by project basis. One goal of isolating the servers from each other is to enable students to focus on various aspects of system administration without interfering with functionality or security of the campus network. An ongoing debate occurs between the faculty, Computer Services and the lab administrator regarding the need to balance security and network functionality with an environment of openness to facilitate learning.

While dividing the servers in this way allows students more hands on system administration opportunities, it can make maintenance of the systems difficult. Addressing server failure is one area that can complicate maintenance. A Jumpstart server that is managed by the lab administrator holds the install configuration and files for all servers. Thus, if a system becomes compromised, it can easily be returned to its initial configuration. Currently, there is not a standard schedule for backups on all of the servers, though it is needed if a reconfigured system is to be returned to its precompromised state. A second area of difficulty is patching the various servers. With better documentation and training, this responsibility could be distributed between the lab administrator and the student assistants. Finally, identifying upgrade possibilities and performance improvement takes a significant amount of the lab administrator's time. Eventually, interested students could be assigned these responsibilities if clear documentation guidelines are developed.

3 System Software Resources

The E420R server is the backbone of the both labs. The operating system used on this server is Solaris 9. The server software for the SunRay terminals is SunRay Server 2.0. By default, the Common Desktop Environment is the window environment. Gnome 2.0 is available on the system, though currently it is not officially supported by Computer Services. All software and user accounts are managed on the E420R server. Tasks that must be performed on the E420R server are account management, user environment configuration, patching, and backups. Overall, these administration tasks take approximately six to ten hours per month.

By far the least time consuming of these tasks is managing user accounts. A script for managing addition and deletion of student accounts was created. Using a text file containing Computer Science course enrollment data, it adds new accounts, resets passwords, locks unused accounts and deletes accounts. By resetting student passwords each semester, one of the most annoying tasks, resetting forgotten passwords, is much less of a problem. The other recurrent issue relates to students who have changed their schedules during the drop/add period at the beginning of a semester. Resolving such situations usually take less than a couple of hours in the first month of the semester.

Until this year, the default Solaris window environment, CDE (Common Desktop Environment) was used. Some of the services that the default environment expects, such

as calendars and email, are turned off to secure the server. Students, therefore, experienced situations where the environment didn't appear to work, for example clicking on an icon and nothing happening. This year the faculty and lab administrator discussed the minimal tools needed by the students and reconfigured the CDE environment to access only available tools. While the research involved in configuring the environment was time consuming, actual implementation was less than an hour. Once the procedure was established, the test server was used to iron out difficulties. It was then applied to the production server. No additional changes to the environment have been required so far this year. As mentioned previously, the Gnome environment is available to users. There has not been time to adequately research proper configuration of this interface, so it is used primarily by advanced students or students with Linux experience.

The E420R server is patched monthly with signed patches from Sun Microsystems. Patching is one of the more time consuming aspects of system maintenance. Before patching the production server, most patches are installed on the test server. This testing of the patches takes between two and six hours depending on the number of patches. The production server can usually be patched in a couple of hours because the effect of the patches has already been tested. Patching almost always involves at least one reboot, so it is generally done on a Sunday morning. Currently, the support servers are patched only at installation. An important improvement to the security and integrity of the lab will be to find a way to regularly patch these servers also. It is expected that this could add up to ten hours of administration time for the lab administrator during times when student assistants are not available to manage those servers.

Backups are done using a DLT8000 tape drive which is connected to the E420R server via SCSI interface. System backups are done monthly, before and after patching, because they involve unmounting file systems. To minimize the amount of time the system is down, these back ups are performed at the same time as system patching. The tapes are stored for one semester or about six months. Home directories are fully backed up once a week and incrementally backed up four times a week. These tapes are kept for one month. The incremental backups are currently handled using a customized script. In order to avoid system down time, snapshots of the user file systems are taken at four a.m. Incremental backups are done from those snapshots. Originally, Veritas software was used for the backups, but, recently, it was decided that the system commands for tape backup would adequately handle back ups. This eliminated the need to purchase additional licensed software.

Typically, the system runs with very few problems. To ensure that problems can quickly be resolved, a Sun service contract for the E420R server and its storage array was purchased. This was useful last semester when a puzzling problem began occurring with input devices freezing sporadically during SunRay sessions. The Sun technician provided diagnostic resources and advice on patches and network configuration. The situation was resolved within one week of the call to Sun. Due to a delay in seeking help from Sun, students encountered the problem through much of the semester. Students found a work around for the problem by interrupting power to the SunRay terminal. It was good that

students were thinking of ways to resolve problems with the system. However, now, students tend to blame the system if something unexpected occurs even when a user action is actually the culprit. It will take some effort to rebuild user confidence in the system.

4 User Software Resources

To be as economical as possible, it was decided to use software distributed under GNU general public or other open source licenses. Both the director of Computer Services and Computer Science faculty had concerns about this. The primary concern for the director of Computer Services involved issues of system security and integrity. While there are no guarantees, steps were taken to minimize likelihood of problems. Because all software runs from one server, the E420R server can be monitored for processes that are destructive or out of control. Ensuring the software does not run with root permissions limits the amount of resources that any one process can use on the server. Monitoring logs, minimizing the processes that are running, closing ports and keeping software updated helped alleviate many of the Computer Services director's concerns. Many of these tasks are automated by the Solaris Security Toolkit ("Solaris Security Toolkit") provided by Sun Microsystems.

Faculty had other worries about open source software. Because UNIX systems are not familiar to most computer users, ease of use came up immediately. With proper documentation, most tools have been easily learned both by faculty and students. Deciding who is responsible for the documentation is one area that is still a bit hazy. The lab administrator cannot maintain proficiency in all the tools installed on the system. Therefore, faculty have had to develop their own documentation for the tools as they use them in class. This has not yet led to difficulties, though at some point a standard documentation might simplify things for both faculty and students. Locating software that would adequately support concepts being taught in courses was also mentioned by faculty. Faculty were asked what tools they would like to have installed on the system. In most cases, either the tools they wanted ran on Solaris or an acceptable alternative was found. In the limited cases where specific Microsoft Windows tools were needed, faculty chose to use the general use computer labs which are all Dell PCs. Finally, faculty wanted students to be able to install and use all tools used in the SITLC and teaching lab on their home computers. In the past, students had to purchase software to do this. An advantage to the open source tools was cost savings for the students. For students with Linux or Mac systems at home, there were no issues because generally software available for Solaris can easily run on Linux and Mac systems also. This left interoperability between open source software and systems running Microsoft Windows as the primary concern.

4.1 General Software

There are some basic user software packages installed on the system (see Table 1). A solution for interoperability concerns was addressed when a tool called CygWin ([CygWin Information and Install](#)) was found. This tool emulates a Linux environment on the Microsoft Windows operating system. It is easily installed and available under the Gnu Public License. All platform dependent software available through the SunRay thin clients also runs well within the CygWin environment. CygWin was installed on all of the Dell machines in the general use computer labs on campus, so students can use any computer lab for homework assignments. For the last two semesters, students in the introductory programming class have been given instructions on how to install and use CygWin it at home. Once installed, students can use the open source tools demonstrated in the SunRay labs on their home personal computers. There have been no reported problems.

<i>Software</i>	<i>Description</i>
CygWin	A Linux-like environment for Windows.
OpenOffice.org	A suite of office tools compatible with most office suites.
Acrobat Reader	A tool for accessing PDF files.
Firefox	The web browser from the Mozilla Internet Suite.
rdesktop	A client for Windows Terminal Services.

Table 1: General Software.

The web browser available to users is Firefox ([Firefox - Rediscover the Web](#)). This is the web browser distributed with the Mozilla Suite of Internet applications. Only the web browser component is installed because, while students need access to the Internet for their course work, there appeared to be no academic advantage to enabling web chatting, email checking, etc. on the SunRay server. At this time, students cannot print from the web browser to minimize the amount of wasted paper. So far, there have been few complaints from students or faculty about this, largely because they can do such printing from personal computers. In the future, it may be necessary to enable printing from the web browser.

Given the common use of PDF files on the Internet, Acrobat Reader ([Adobe Reader](#)) is a requirement for any computer. It installed easily onto the Solaris environment. It integrates seamlessly with Firefox using a plugin that comes with the SPARC Solaris distribution of the reader. The PDF files load quickly within the web browser. If students need to print PDF files, PDF files can be printed by opening the PDF file directly in Acrobat Reader.

OpenOffice.org (OpenOffice.org) is a complete suite of office applications. At this time the database tool was not installed. Instead, the database server provides access to a complete database system. There were concerns about the network and system load that this application might generate. It has caused no problems so far, though it is not yet widely used by students. OpenOffice.org integrated more quickly and easily with the Gnome environment than CDE. No significant problems have been reported in either environment. Because OpenOffice can both open MSOffice files and save in MSOffice formats, there have been few difficulties across platforms. Most frequently, fonts available with one application are not available in another. This is not considered a significant issue by any users.

rdesktop (Chapman) is a free client for Windows Terminal Services. When the instructor for a network security course suddenly changed, rdesktop was installed to allow students to access a Terminal Services server which provides them access to an isolated Cisco network lab. Because it runs only under the individual user id, there were no additional security steps needed on the SunRay server. The instructor of the security network class assumed responsibility for securing the Terminal Services server. Only about ten students need to run this client at any given time, so it has had a minimal impact on server performance. If more courses use it, there may be a larger impact on the SunRay server performance.

4.2 Development Tools

<i>Software</i>	<i>Description</i>
jEdit	Java-based code editor for multiple programming languages.
GNU Compiler tools	Compiling tools for C, C++, and Fortran.
DrScheme	IDE for Scheme programming.
NetBeans	IDE for Java programming.
Python	Tools for compiling Python code.
DotGNU Portable.NET	Tools for compiling C# code and running .Net applications on UNIX systems.

Table 2: Development Tools.

The SMSU-Schwans IT Learning Center provides students access to a number of programming languages (see Table 2). C++, Java, Python, Scheme, Fortran and C# are all currently available for student and faculty use. Because all of these development tools either work within CygWin or are Java based, students could easily and cheaply recreate

the development environment on most home personal computers. This section briefly describes the development tools available.

To simplify code creation for the languages, it was decided to use a single programmer's text editor called jEdit (Pestov). Because it is written in Java, jEdit works well with both UNIX and Windows systems. It has been installed on computers in all of the labs on campus. Additionally, students can easily download it at home. During their first programming course, students are given directions for downloading and installing the software on their home computers. Plugins for beautifying code and syntax checking are added as needed by user from the jEdit website. One issue that has come up occurs when students log out of their thin client session without closing jEdit. This leaves a process hanging on the server. Currently, students are asked to be sure to close all open applications at log out, but a better solution will need to be found.

The C++ development tools came from the Gnu Compiler Collection ([GCC Home Page](#)). A few of the command line development tools this collection provides are C/C++, Fortran and Java development. This was the set of tools that required the CygWin environment so that students can do their C++ homework assignments on Windows machines. Fall semester of 2004 was the first time that faculty used these tools in a beginning programming course. Faculty feedback to the lab administrator was that they were able to move into programming concepts much more quickly than when an integrated development environment was used. Student feedback, especially from advanced students having to relearn the compiling process, bemoaned the loss of a graphical debugger. Unfortunately, a graphical debugger for C++ has not yet been located.

The Gnu Compiler Collection was acquired from the Sunfreeware.com ([Freeware for Solaris](#)). This site offers free software packaged for the Solaris OS. The advantage to a site like this are software packages that can be quickly verified and installed. While precompiled, packaged software is quick to install, it limits the install location and can end up installing multiple versions of libraries. When open source software is compiled from source code, usually, fewer different versions of libraries are needed. Sunfreeware.com provides both precompiled, packaged software and source code for the software. It definitely makes locating software for Solaris easier.

Two integrated development environments, NetBeans ([Welcome to NetBeans](#)) and DrScheme ([DrScheme](#)), are available to students. NetBeans currently is configured only for Java development, though it can be configured to support C++ and XML development. If a graphical debugger for C++ cannot be found, the C++ plugin for NetBeans will be installed. DrScheme is the graphical development environment for Scheme programming. Both environments were simple to install on the SunRay server. Students have used them in both the Gnome and CDE window environments with no reported problems.

The most recently installed development tools are Python (Python Programming Language) and DotGnu Portable.NET (Southern Storm Software, Pty Ltd.). Both proved slightly more difficult to install on the SunRay server because they each needed to be compiled from source code. Python required a specific version of the tar utility that is not part of the standard Solaris installation. Portable.NET relied on numerous utilities that had not been installed on the SunRay server. In order to avoid cluttering up the SunRay server with these tools that may not be needed again, a SunBlade workstation was set up with the required utilities and libraries. Because the SunBlade and E420R have the same architecture and operating system, code compiled on one will run cleanly on the other. Once the code is compiled, it can be consolidated into a package that is simple to install on both the test and production environments. When planning to use open source software that must be compiled from source, it is important to allow for up to a month of research and development time before having the production version of the software ready.

4.3 Course Specific Software

One of the advantages to the SMSU-Schwans IT Learning Center is the flexibility offered for use of a variety of course specific applications (see Table 3). Examples of these include TKGate (Hansen), Marie ("Marie Simulator Software"), and Automata Simulators (Head). In the past such applications were not uniformly installed on campus-wide computers because the headache of locating, installing, and maintaining them was not justified by their very limited use. With the current system, faculty can request that certain applications be installed for a particular course. Given enough lead time, the application can be researched, installed and, ideally, documented before the semester starts. Because users are always on the same system, faculty can demonstrate these tools in the teaching lab exactly the same way that students will access them when working on homework. Most of these applications can easily be installed on their home computers as well.

<i>Software</i>	<i>Description</i>
TKGate	Graphical digital circuit simulator.
Marie	Java-based machine simulator.
Automata Simulators	Java-based automata simulators for FSA, PDA, and Turing Machines.

Table 3: Course Specific Software.

An additional benefit of this integrated system is ease of managing course related documents including homework. Each course has its own space reserved on the system. A small shell script enables students to submit homework to a folder in the course

directory. Faculty can then easily review the homework with the exact same tools the students used while completing the assignment. The course directory also has a public area where faculty can place documents for students to access anytime. This may become obsolete by Desire2Learn as more faculty use that resource, but it served its purpose during the transition.

4.4 Summary of User Software Resources

Overall, the use of these open source applications has worked well. All applications perform their functions reliably and have been available on both UNIX and Windows systems. The cost to both the department and the students for software licenses has definitely decreased. However, more time must be spent researching the software, security issues and dependencies. Added administrative overhead will occur when updates to the various packages will need to be tracked down. Currently, the plan is to research the updates at the end of each spring semester. Over summer break the system will be brought down and completely reinstalled. Time will tell if this approach will prove too time consuming.

5 Future Plans

There are many ways to improve the SMSU-Schwans IT Learning Center and teaching lab. The general category of software improvements include some additional development tools. In addition to the previously mentioned graphical debugger for C++, some type of versioning software should also be available. One possibility is the use of NetBeans to satisfy both needs. Another aspect of software improvements is in the area of maintaining patches and updates to the open source applications. While better documentation will help streamline the process, it has the potential to require more of the lab administrators time.

Some additions and changes to the support servers will also improve the usefulness of both labs. Setting up a Terminal Services server to access Windows development tools via rdesktop would allow any Computer Science course to be taught in the teaching lab. This would make it easier for Computer Science faculty to schedule class lab time. The current configuration using a workstation for the database server is not the best use of equipment. There may not be enough processing power if use increases. Also, a web server needs to be available for students to explore the complexities of web development. Therefore, next year, one of the Cobalt servers will be configured to host a combined web and database server.

Currently, Computer Science students need to manage at least two separate logins and passwords. While this is hardly a burden, there should be a way to authenticate users across all campus systems. Integrating the SunRay login process with an Active

Directory server that currently manages all student accounts would consolidate the login data. Another area that could be integrated with existing Computer Services resources is the use of a Dell system that automatically backs up all servers except the E420R. This would free up some of the lab administrators time as well as ensuring that the backups occur uniformly on all Computer Services servers.

6 Closing

Of the many questions raised at the beginning of this paper, some answers have been found in the process of configuring and maintaining this dedicated computer lab. Balancing the requirements of Computer Services with the needs of the Computer Science faculty can be done. This balance can occur in part because the consolidation of all student and faculty work runs onto one server. Thus, any problems can be traced to that server which can be isolated. Integrating the SITLC with Microsoft products has been more difficult. Commitment from all parties will go along way toward achieving this goal. Yet, one wonders are there other options? Other schools have taken different approaches. Belles and Miller invested in multiple swappable hard drives which could be configured per course and student within course. Sloan created a layered architecture in the remotely accessible lab used for teaching networking courses. How do these configurations evolve over time?

The issues raised by Computer Science faculty were very different and often conflicted with concerns Computer Services people identified. No surprise to anyone, this shows that the teaching of Computer Science has very different needs than use of computer technology to support academic programs. Authors like Rahman et. al., working at time when computer technology was first moving out of the department of Computer Science, focus on serving the computing needs of an entire educational community. Interestingly, things have come full circle now. Many Computer Science departments, especially those with limited resources, are looking for ways to reintroduce tools and a sense of community that was lost when mainframe systems went away (Wooley, p. 229).

Somewhere between Computer Services and the Computer Science faculty are the students. While the SITLC is planning for ways to involve students in system administration opportunities, poor documentation of system administration tasks and limited student time need to be overcome to fully use these opportunities. How to go about improving this is an area in need of exploration. Is this a responsibility of a dedicated lab administrator or is it a curricular issue that needs to be addressed from the faculty? Could students themselves find the way to make these resources more fully utilized?

Finally, one of the biggest questions left is what is missing? What has been overlooked? This is where the author hopes to join the dialog about computing technology in Computer Science education.

References

- Adobe Reader. 2005. Adobe Systems Inc. 10 Mar. 2005.
<<http://www.adobe.com/products/acrobat/readermain.html>>.
- Belles, Robert & Miller, Jacob R. Multi-purpose Specialty Labs: Providing Specialized Computer Labs Both Cost and Space Effectively. presented at CCSC: Easter Conference. 2001. Consortium for Computing in Small Colleges.
- Chapman, Matthew. rdesktop: A Remote Desktop Protocol Client. no date. Open Source Technology Group. 10 Mar. 2005. <rdesktop.org>.
- CygWin Information and Install. 2005. Red Hat, Inc. 10 Mar. 2005.
<<http://cygwin.com/>>.
- DrScheme. no date. PLT 10 Mar. 2005. <<http://drscheme.org/>>.
- Firefox - Rediscover the Web. 2005. The Mozilla Organization. 10 Mar. 2005.
<<http://www.mozilla.org/products/firefox/>>.
- Freeware for Solaris. 10 Mar. 2005. Steven M. Christensen and Associates, Inc. 10 Mar. 2005. <<http://sunfreeware.com/>>.
- GCC Home Page. 28 Feb. 2005. Free Software Foundation, Inc. 10 Mar. 2005.
<<http://www.gnu.org/software/gcc/>>.
- Hansen, Jeffery P. TKGate Home Page. 18 Mar. 2004. Open Source Technology Group. 10 Mar. 2005. <<http://tkgate.org/>>.
- Head, Eileen. "Formal Languages and Automata Theory." Tools. no date. Binghamtom University. 10 Mar. 2005. <<http://cs.binghamton.edu/~software/>>.
- "Marie Simulator Software". Student Resources - Essentials of Computer Organization and Architecture. 2003. Jones and Bartlett Publishers. 10 Mar. 2005.
<http://computerscience.jbpub.com/ecoa/student_resources.cfm>.
- OpenOffice.org: Home Page. no date. Sun Microsystems, Inc. 10 Mar. 2005.
<<http://www.openoffice.org/>>.
- Pestov, Slava. jEdit - Programmer's Text Editor. 31 Dec. 2004. Open Source Technology Group. 10 Mar. 2005. <<http://jedit.org/>>.
- Python Programming Language. 2005. Python Software Foundation. 10 Mar. 2005.

- <<http://python.org/>>.
- Rahman, Abdul; Bidin, Haji; Drabble, Gwynneth. "An Organizational Approach Towards the Development of Educational Computing in a University Environment." Computer Education. 1990. Pergamon Press. v14, n2. pp. 137-143.
- Sloan, Joseph D. "A Remotely Accessible Networking Laboratory." JCSC. 2002. Consortium for Computing in Small Colleges. v18, n2. pp. 215-222.
- "Solaris Security Toolkit ("jass")." Sun BluePrints: Scripts and Tools. 15 Nov. 2004. Sun Microsystems, Inc. 10 Mar. 2005. <<http://www.sun.com/blueprints/tools/>>.
- Southern Storm Software, Pty Ltd. 16 Jan. 2005. Southern Storm Software, Pty Ltd. 10 Mar. 2005. <http://www.southern-storm.com.au/portable_net.html>.
- Welcome to NetBeans. no date. Sun Microsystems, Inc. 10 Mar. 2005. <<http://www.netbeans.org/>>.
- Wooley, Bruce. Utilizing a Computer Lab to Improve Retention and Recruiting of Computer Science and Computer Information Science Students. presented at CCSC: Easter Conference. 2002. Consortium for Computing in Small Colleges.