Electronic Submission of Undergraduate Thesis: Modification of the University Of Virginia Electronic Thesis Submittal System

A Thesis in TCC 402

Presented to

The Faculty of the
School of Engineering and Applied Science
University of Virginia

In Partial Fulfillment
of the Requirements for the Degree

Bachelor of Science in Computer Science

by

Zachary D. Connelly

March 24, 1998

On my honor as a University student, on this assignment I have neither given nor received unauthorized aid as defined by the Honor Guidelines for Papers in Technology, Culture, and Communications Courses.

_____

Approved _____
James C. French, Technical Advisor

Approved _____
Edmund P. Russell, III, TCC Advisor

Preface

Beginning my final year here at the University of Virginia, I was unsure of what my fourth year thesis was going to be. A friend of mine mentioned to me a project that Professor James French was running concerning the submission of fourth year theses in electronic format. I was unfamiliar with many of the concepts involved in the project, but I was intrigued to do a project that would not only be used, but would impact, the students at UVA. I am grateful that Professor French gave me the opportunity to participate in this project.

I have been aided in this project by many people whom I wish to thank. Cristina Sharretts and Jackie Shieh are the two librarians I have had contact with. They have helped me understand their perspective on what this project must entail for it to be successful. I have learned a great deal about the background details on how the library works.

I would also like to thank graduate student Allison Powell. She has been my Perl guru who has helped me understand this powerful but confusing language. I am sure she has been very busy, but she always took the time to e-mail an answer or see me personally.

This is the largest report I have had to write on my own. Thankfully, I have had a few people help me in this writing by editing my paper. Thank you Jeanine Lee and Ingrid Neuland for patiently reading rough drafts for this thesis.

This project has given me the opportunity to do work that people are counting on me to finish. The work I do will exist for many years to come. I am sure it will be improved repeatedly once I am gone, but at least the foundation I have laid will still be around. This has been a great opportunity for me and I will take a great deal of experience away from it.

Table of Contents

List of Figures

Glossary and Acronyms

Acrobat PDFWriter – This is the software that a student uses to convert a document file into a PDF file. PDF is the file type used by the library to display the thesis over the Internet.

boolean – mathematical logic with the only values being "True" or "False"

DIENST - Distributed Interactive Extensible Network Server for Techreports

ETD – Electronic Thesis and Dissertations

GUI – Graphical User Interface

ID Number – The student's Social Security Number

MARC record – Machine Readable Catalog record; a text file used by the library to enter a new material's bibliographic information into the library's database.

NCSTRL - The Networked Computer Science Technical Reports Library

PDF – Portable Document Form

Perl – Practical Extraction and Report Language

SEAS – School of Engineering and Applied Science

TCC – Technology, Culture, and Communication; this is the department in the Engineering school that coordinates the thesis project as well as teach humanities to all engineering students.

Usability – A combination of user-oriented characteristics such as ease of learning, high speed of user task performance, low user error rate, subjective user satisfaction, and user retention over time.

UVA – University of Virginia

VT – Virginia Tech

Abstract

As more people gain access to the Internet, the ability to obtain information from different sources around the world will increase. To incorporate the University of Virginia's student body in this information revolution, an electronic publishing system was developed here for use by fourth year engineering students for their theses. After the pilot project was completed and successful, it was important that the system be modified for increased utility, both for the students and the library. This new system must eliminate many of the sources for user error and quicken the incorporation of theses into the library's computer system. This project's focus is to achieve these goals for use by the engineering class of 1998.

Chapter 1 – Introduction and Literature Review

With the advent of the public Internet in the early nineties, information is readily available to a larger community of people at a much faster rate of transport than in any time in history. The job of people in the information technology field is to develop ways to organize information logically and move that information as efficiently and quickly as possible. This thesis, the continuation of the University of Virginia (UVA) School of Engineering and Applied Science (SEAS) Undergraduate Theses Project, tries to accomplish both of these goals. This project will attempt to streamline the current process, making thesis submission easier, quicker, and less error-prone for students, faculty, and the library staff by further automating the submittal process.

1-1 Problem Definition

1-1-1 Context

Initial discussions for digital libraries began in the late 1980s. People recognized the magnitude of information being produced and needed an efficient means of storing and retrieving that information. The first steps toward storing theses and dissertations on-line began at Virginia Tech with its development of the Electronic Thesis and Dissertation Project (Fox et al. 1996). Since then, there have been collaborations between institutions and industry to create efficient and robust digital libraries, most of which hold theses, dissertations, and academic papers. One example of this system is The Distributed Interactive Extensible Network Server for Techreports (Dienst 1996), the first protocol based on HTTP, and The Networked Computer Science Technical Reports Library (NCSTRL 1996), a distributed system providing researchers access to computer science departments around the world through the Internet.

Although research into digital libraries exists today, it is important this project continues at the University of Virginia because of the benefits it will provide to students. This system will

introduce students to electronic publishing and the use of digital libraries for their documents. It will improve library services by offering timely and complete access to undergraduate theses via a networked, full-text searchable archive. The knowledge and experience from this project can also be applied to extend the program to graduate theses and dissertations in the future.

1-1-2 Concepts

Here is the submission scenario for students to submit their theses electronically based on the old submission process:

1. The student writes a thesis in whatever word processing program that is comfortable, on any platform, and includes any graphs, charts, diagrams, etc.

2. When the thesis is completed, the student instructs the print command on the word processor to create a Portable Document Format (PDF) file using Adobe software instead of a paper copy.

3. The student accesses the WWW homepage for electronic submission and follows the directions for submittal.

4. The student fills out the corresponding form on the web page (name, advisor, title, etc.) and submits the thesis.

5. The PDF version of the thesis is uploaded from the student's computer to the library archive.

6. The library confirms the acquisition of the thesis and sends an e-mail verifying a successful transmission.

7. The library adds the thesis to the library database if it passes qualification (by TCC advisor, technical advisor).

8. All the theses are available for anyone to read over the Internet from UVA's VIRGO system.

This is the current system. The focus of this thesis is to change steps four and seven. The submittal form will be altered to minimize user error by using a constant list of people instead of having the student enter in the data. This will minimize misspellings, common names dissimilar

to official names, and provide some sort of security. Step seven will be changed by converting the submitted information into a format the library can use for automatic cataloguing.

1-2 Literature Review

Although digital libraries are a new concept, research in the field has been very extensive. The first work with Electronic Theses and Dissertations (ETD) began in 1987 with a meeting between representatives from the University of Michigan, ArborText, SoftQuad, and Virginia Tech (Fox et al. 1996). Work on the first system began soon after that, converting pre-existing theses and dissertations on diskette to the Internet. Soon, many other projects followed, including the Coalition for Networked Information in 1992, the Monticello Electronic Library in 1993, and finally the federally funded National Digital Library of Theses and Dissertations, pioneered by Virginia Tech and involving multiple universities across the country, including the University of Virginia (Fox et al. 1996).

The current system here at The University is based on one system the Distributed Interactive Extensible Network Server for Techreports (DIENST). DIENST is a distributed system providing a single point of access to research results from distributed document collections. This system holds the actual information in collections. The digital library servers provide three services: repository services that store and provide access to documents; index services that allow searches over bibliographic records; and user interface services that provide the human front-end for the other services (NCSTRL 1996).

The DIENST protocol is based on the World Wide Web HTTP protocols. The main contributions of DIENST and the server include an extensible document model with an object-oriented interface, an HTML forms-based interface supporting search queries, and the ability to perform parallel searching of multiple document indices (Dienst 1996). The DIENST protocols are the technological backbone for the SEAS Undergraduate Thesis Pilot system.

Another facet of this project deals with Graphical User Interfaces (GUIs). GUIs are the means by which the user enters data into the computer. In the beginning, interfaces were purely text-based, like the MS-DOS and UNIX operating systems. Beginning with the Apple operating system and followed by the Windows system, GUIs moved away from command-line and menu-driven interfaces to ones the user could directly manipulate objects represented as icons (Fowler, 1995). It is with this transition that the usability of products became important. Usability is defined as "a combination of user-oriented characteristics such as ease of learning, high speed of user task performance, low user error rate, subjective user satisfaction, and user retention over time" (Hix 1993). These characteristics are important for this project because students will not spend the time to submit their theses if the task is too difficult or time consuming. The whole process must be straightforward, simple, and fast in order to attract the volume of students anticipated for this system.

The programming in this project will be done in one main language call the Practical Extraction and Report Language (Perl). Perl began as a general-purpose tool, used by creator Larry Wall, to produce reports on a file system that he was using (Schwartz 1). Perl is designed to assist the user with common tasks in UNIX. Its main strengths lie in string manipulation functions that are inherent in the language. It is also a good language to create HTML pages that include forms. These forms allow the user to enter in information that the program sends to different servers for processing.

1-3 Rationale for and Scope of the Project

The goals for this project are

- To modify the current system for the Fall TCC 402 class for use in submitting theses.
- To garner information from testing to make needed improvements to the system.
- To have the finished system completed for use by the Spring TCC 402 class.
- To have over ninety percent of the volunteer students be able to submit their theses successfully without complications.

This project, coordinated by the library administration, is intended to be the first step in the process of eliminating bound copies of theses in the library. If it can be proven that this system is robust and correct enough for all the relevant parties (TCC faculty, technical advisors, the students, and the library), then it is my hope that electronic submission will become the standard, and not just a voluntary, option.

1-4 Overview of contents for the rest of the report

The rest of this report clearly explains the process by which this project was completed. The main parts of the report follow the goals set forth in the preceding section (1-3, Rationale).

The second chapter gives a synopsis on the history of the UVA SEAS Submission Project and explain how a thesis is submitted to the library. The third chapter discusses my preparation for the project. This includes learning how the system worked, running some tests to check the validity of the PDF software, and raising awareness of the project among the Fall TCC students. The fourth chapter explains the work done in the fall to prepare for the TCC 402 class. This included designing an exit survey form to gather user information on the entire submittal process; the testing methods employed to get user information by observing students actually submitting their theses; and using the data to determine what changes need to be made to the submittal form.

The fifth chapter discusses the main body of the report, the redesign of the submittal form. This chapter explains the fundamental changes that were employed to minimize user error by adding key features to the form. Some of these features include adding pre-existing information behind the form to guarantee security and the correct spelling of names, as well as a program to convert the gathered information into something useful for the library. The sixth chapter discusses some of the results of the project. This includes completed results and projected estimates on the number of people expected to use the system. The final chapter summarizes the work done for my thesis and attempts to put it into perspective. This entails analyzing some of the positive ideas that came out of the process as well as some of the problems that arose. It also discusses some future work that might be done to make the project even more effective.

Chapter 2 – History and Synopsis of the Project

This pilot program was a thesis project done by a graduated engineering student, Mariahna Moore. Her thesis (Moore 1997) involved successfully taking a pre-existing electronic submission system and modifying it for UVA's purposes. It is important to discus her system first (this system can be found on the Internet at http://viva.lib.virginia.edu/etd/cgi-bin/etd.html). In the process, I can give a detailed description of the entire system. This includes how a submission goes from the student to its final location in the library's database.

2-1 Previous system

Mariahna Moore based her system on the DIENST system, modifying Virginia Tech's (VT) electronic submission system. Her system took out some of the complexity of VT's system, namely some availability options and the ability to have multiple advisors. Her system was very logical and efficient. The student would first create a PDF file in the traditional manner. The student would then go to the form site and fill in all the information needed. This information would be entered through text boxes instead of scroll boxes. Because of this, students could enter in names that were not the same as formal names. It also allowed the student to enter in names that were different from the required form she requested (the form wanted last-name-first; a few people entered their names first-name-first). The student also had to check a box showing they understood the copyright laws and that a title page approving the thesis was turned into TCC. The purpose of Miss Moore's thesis was to show that an ETD system could be implemented at UVA. However, many parts of the system need improvement. This improvement is the bulk of my thesis.

2-2 How a thesis gets from the student to the library

There are six steps that must be followed before a thesis is put into the library.

1. The student first creates a PDF file for the system using Acrobats PDF Writer.

2. The student enters the necessary information into the form. This includes all the bibliographical information as well as the location of the PDF file.

3. After reviewing the information, the student sends all the information over the Internet to the library. This information is stored into a directory waiting to be catalogued by the library.

4. A librarian accesses the bibliographical information and makes sure the PDF file is not corrupted. If everything is OK, it will be approved. If there is a problem, it will be rejected and an e-mail is sent out to the student. The student will most likely correct the problem and resubmit the thesis again.

5. If it is approved, the librarian gives the document a call number and moves the thesis to its permanent directory for storage.

6. The librarian then creates a MARC record for the thesis. This is used to enter the bibliographic information into the library's computer system. This was done by hand using the information sent by the student. After the record is complete, the thesis is on the system and viewable over the Internet through the VIRGO system.

The focus of this thesis is to modify steps two and six while leaving all of the background network code intact.

Chapter 3 – Phase I

The first phase of the project prepared the system for the Fall TCC 402 class. This entailed learning the mechanics of the system and running some tests to see what kinds of applications might not work with the Adobe PDF Writer software. I also spent time advertising and promoting the system to the students in the TCC 402 class.

3-1 Preparing the system

There were two main requirements in preparing the system for use. The first was to learn the system so I could help people who might need assistance. I spent time playing with the system: going through the whole submission process, trying to cause errors in the system by incorrectly using it, and trying to think about some possible long-term changes I could make after Phase I. I also spent some time looking at the underlying code to see exactly how the program worked. However, at the time I was relatively unfamiliar with the code, so this part was more of a chance to see the code, not analyze it thoroughly.

The other requirement for preparing the system was to test if there were any applications that the Adobe software could not convert into PDF format. I tested the system by creating documents that contained various applications imbedded in them. These included screen images, graphics packages such as Mentor Graphics, and all the various word processors (Microsoft Word, Corel WordPerfect, and basic text files). The software converted all of these applications correctly.

3-2 Awareness

Because this pilot program is being done on a voluntary basis, it was my job to make the Fall TCC 402 class aware of their options with regard to electronic publishing. With the help of the library (specifically, Cristina Sharretts), we attempted to inform the students what we were trying to do with the pilot program. We went into class to give them instructions on how to use the system. I also sent e-mail out to all the students asking for any problems they had. The students asked some of the questions, most of them concerning the conversion process. One student asked how she could take her files, which were created on an Apple platform, and port them into a Word document on a PC platform. Most said they anticipated no specific problems and would not have any problem with the submission.

Chapter 4 – Phase II

This section of the thesis concerns the actual testing done on the Fall TCC 402 class as well as the analysis that came out of it. This focused on developing the instructions on how to use the system, creating the necessary documents needed for attaining feedback from the users, and then analyzing the results to incorporate necessary changes in the new form being developed.

4-1 Creating Instructions

Although the system is mostly self-explanatory, it was necessary for a set of instructions to be created to give the students something tangible to follow in case they had trouble with the submission. The most important part of the instructions was getting users to create a PDF files correctly and getting the user to the submission form. Having done an exercise in instruction preparation in one of my classes earlier, I am aware of how difficult it is to create written instructions that are helpful and not hindering. I went through the process many times, writing down step-by-step what I was doing. I had a mind set not of an expert in using the system but of a novice. I tried to make each instruction straightforward and as short as possible. After I went through a few drafts of the instructions, I had some colleagues try the system using my instructions, without giving them any help – just observing. I used their feedback as well as observations I had made to modify the instructions where necessary (a copy of the instructions is located in Appendix A).

4-2 Employing Testing Methods

After students completed their final theses and had them approved by both their TCC and Technical advisors, they were asked to submit them electronically (currently the submission program is on a voluntary basis). Using testing techniques learned in a previous class, I asked volunteers if I could watch them as they submitted their thesis. With this direct observation, I

could tell exactly where students had a problem submitting their thesis. I also attained information on exactly what that person was thinking as he or she progressed through the process. It was a great tool to see where trouble spots in the system and/or the instructions were. In addition, I could aid them in the process if they get completely frustrated with the process. Unfortunately, due to time constraints for the graduating students, I was only able to use the direct observation technique on two students. However, these two students made very few mistakes overall.

Another part of the testing process was for users to fill out an exit survey about the process. Four students gave me feedback information about their experience with the system. The survey (a copy of which is found in Appendix B) included background questions, such as what word processor was used, and questions directly related to their experience, such as what problems, if any, they had and what suggestions they have to make the process easier. Most of the surveys reported very little trouble with the system, with only some cosmetic changes suggested (like having a reset button on the first form so that students could delete the information in the form when they were finished with it).

4-3 Discuss results and plan for the spring

Using all the information garnered from the observations and the surveys, Professor French and I mapped out a plan for the spring semester. We included some of the requests made by students that were constructive (such as the reset button), but most of our focus was in eliminating possible errors between the library and the student. One of the main problems the library has is when students use nicknames for themselves or their professors. Instead of having the student type in a name and an advisor's name, the student chooses these names from lists. The library provided these lists (after a failed attempt to receive them from the Registrar's office). The form also has a new ID field to check the ID number entered by the student against one on record

with the library. These changes eliminated many of the possible places where students entered information that is correct by their standards, but incorrect by library standards.

Chapter 5 – Phase III

This was the spring phase of the project. After discussing what changes needed to be made to the system, this phase accomplished those changes. There were five main objectives in this phase:

- Redesign the submission form
- Develop the script that creates a MARC record for the thesis
- User test the new submittal process and make any needed adjustments
- Move all critical files to the libraries server
- Promote the new submittal process

The work discussed was done in time for the fourth-year engineers to use after their thesis was submitted and approved by both their advisors. The explanation of work that follows will contain the steps used to solve the problems in the system. For a more detailed explanation of the code created for the system, please see Appendix C. For a screen shot of the new form, please see Appendix D or go to the site on the Internet at:

http://viva.lib.virginia.edu/etd/cgi-bin/etd.html.

5-1 Redesign the submission form

One of the major goals for Phase III was to change the submission form. These changes come directly from the analysis done in Phase II. There were four main changes done to the submission form:

- Change the text boxes of the technical advisor and the student name into scroll boxes
- The addition of an ID Number and the facilities to check it against an ID Number on record with the library
- The addition of a new TCC Advisor scroll box
- The addition of a reset button to erase the information after submission

To prepare for these changes, I copied all the relevant files into a new directory. This was done so that an existing copy of the old form existed in case of a computer disaster or a failure on my part to get the new submission form to work.

To bypass the problem of students entering in incorrect names for themselves or their professors, Professor French and I decided to have the student choose the names from lists. The first task was to obtain a list of faculty and a list of students and their Social Security Numbers (called ID Numbers). The library provided text files containing the information they use for their own library services. The next step incorporated the information in these files into the script. Using Perl's file mechanisms, the names were read from the file and inserted into the corresponding scroll boxes. Using the CGI mechanism "scroll box", it handled all the necessary background information to create the box; I only needed to provide it the values that go into the box.

The addition of the ID Number required (1) the creation of a text field on the form and (2) a checking mechanism to check the entered number against the one the library has for the user. The file that included all the students' names also included the corresponding ID Numbers. After the student name has been chosen, the ID Number for that person is also taken from the file. It is then compared against the ID number submitted by the user. If the numbers are not equal, a message prompts the user to go back and change it. The user will not be able to submit the thesis if the numbers are not correct. This mechanism makes sure no one can submit a thesis under someone else's name.

Because the library cataloguers require both advisors' names in their records, a new TCC Advisor scroll box has been created. It is similar in nature to that of the faculty list – the professors in TCC are listed in a file and the names are imported into the scroll box. This allows the library to change names quickly by changing the file and not the actual Perl code.

In the user tests done in Phase II, one user suggested adding a reset button to allow users to erase their information after completing their submission. This allows the student to erase any information that might be personal in nature (i.e. the ID Number) and would not like other people

the chance to see. Because it was an easy addition, I created a reset button on the bottom of the submission form.

5-2 Develop the script that creates a MARC record for the thesis

To make additions to the library's catalogue of materials, a text file must be created that contains all the necessary biographical and clerical information for the system. This text file is called a MARC record. Because most of the information that goes into a MARC record will already be entered with the submission form, the library wanted to have the system generate a MARC record, applying as much information from the submission as possible. After reviewing a typical MARC, it became evident that this was possible and many of the needed fields could come from information entered by the user.

The first problem to consider was when this record should be generated. As explained in the synopsis (Chapter 2), there is a separate mechanism for the library to accept the thesis and move it to the upload directory. The best time to create the record is at this point since it is assumed that the library would not accept the thesis if it was not ready. Therefore, the record generation is a function, called by the script at the appropriate time, that created the file.

I obtained an example of a typical MARC record used for the fourth year theses that are on-line (all the following line numbers are from the example in Appendix E). The Perl function is a mixture of information that remained constant, such as lines 516 and 538, and lines that contained specific information to each thesis, such as lines 100, 245, and 590. The specific information is directly taken from the CGI variable that contains all the values. The only lines that the library must enter by hand are the ID number (located in line 008) and the two Internet addresses for the abstract and the full PDF document (line 856).

5-3 User test new submittal process and make necessary adjustments

After making any significant changes to an interface, it must be tested to make sure that (1) all the new features added are correct and the user understands them and (2) the entire program is better than it was before. One way to do this was to user-test the new system in a similar fashion as in Phase II. It was important that the testers were people who would actually use the system. Therefore, I got a few of my engineering friends to look at the system and try submitting a dummy document correctly. I did my best to get people who were in different disciplines of engineering because some people who are submitting their thesis will have less experience with computers than others. For example, Computer Science majors typically would have an easier time submitting than Civil Engineers only because Computer Science majors have more experience with the tools needed for submitting.

I tested three people after completing the changes to the form. Two of the users were Computer Science (CS) majors, the other was a Civil Engineering major. Both CS majors had no trouble submitting a dummy thesis. They both agreed that the new method was very easy to use. Their only complaint was the technical advisor's scroll box contained so many names it was difficult to go directly to their advisor. They suggested having a text box to type the name in and the scroll box would compare the typed name with those in the scroll box and find the name closest in spelling. This method is commonly used in other applications. The Civil Engineer took more time with the submission, but he to did not have any trouble either.

5-4 Move files to the library server

After completing the new form, creating the MARC record function, and making any other necessary changes, the files needed to be moved from the CS servers to library servers. Since the system is a library tool run by the librarians, it was logical to move the system to a library server so they would have direct control. This required me to use a File Transfer Protocol program to

move the files from one server to another. Because the software is directly dependent on which server is being used and which directory the files are in, it was important to make changes to some of the directory paths. After moving the files, I did some test runs of the system to make sure all the different aspects of the program functioned correctly.

5-5 Promote the new system

Because the system is in a trial status, any submissions made by students are purely voluntary. It is important to the success of the project that a high percentage of fourth-year engineers use the system. While most of the advertising and promotion of the electronic submission project will be done by the library and Professor French, I felt it was important that I tell as many as my friends about the new project. I must do a good job in my TCC class presentation of my thesis to make people aware of the project and persuade them that electronically submitting their theses is a benefit to them. Hopefully by my telling friends and colleagues about the project, many engineers will use the system. Unfortunately, the exact number of students who use the system will not be known until after this thesis has been turned in.

Chapter 6 - Results

6-1 Expected Results by Due Date (March 24)

By the due date of the thesis, I have accomplished four main tasks for the project. These include:

- Completion of the new submission form. It will include selection boxes to choose certain fields and validation checks. These checks include verifying the correct ID number for each student and making sure all fields have values.
- A Perl script (a program written in the computer language Perl) that translates the information submitted by the student into a library catalogue record (referred to as a MARC record).
- Transferred all the necessary files and programs from the server located in the Computer Science department to the designated library server.
- User-test the new system with both experienced and inexperienced computer users.
- Satisfactorily promoted the use of electronic theses submission to the fourth-year engineering students. This will be somewhat subjective, but the effectiveness of the promotion can be directly seen in the percentage of people who use the new process.

Determining whether a program has been completed is very subjective. Any program can be improved by making it more efficient, more usable, or by including more features. My project is complete if the design goals set out in my proposal are finished and any further needs the library or my technical advisor requests are completed.

6-2 Expected Results after Due Date (March 24)

Unfortunately, most of the results for this project will occur after the thesis has been submitted. Because of the nature of the project, submission statistics of the fourth year engineering students will be one of the most critical results. The results will represent:

- How many total people submitted their theses
- How many people decided not to submit their theses
- How many people had some kind of difficulty submitting their thesis, either technically (Acrobat would not convert their thesis) or because of poor usability (students could not maneuver through the submission form)

Although these results will not directly affect the written thesis, the library can improve the system and show that the process is a viable option with the information. The information can also be used in the oral report.

Chapter 7 – Conclusion

As stated in the Results section, participation results will not be known until students are ready to submit their final binder. I have thoroughly tested the system, submitting test documents to see if the different parts of the total program work in preparation for the submission period. This included the actual submission form, the acceptance of the thesis by the library, and the translation of the thesis from the submission form into a MARC record. I have also performed user tests on the system at every periodic improvement.

7-1 Significance

The Electronic Thesis and Dissertation (ETD) project is a very important step for the University to grow as a technologically advanced school. It will benefit students, the library, and the University as a whole. Students will be able to experience electronic publishing as well as make the whole submission process quicker and cheaper. The library will get an electronic means of cataloguing documents, saving both time and space on the shelves. The University will gain positive exposure for having an electronic thesis submission mechanism, and the system will allow students a larger environment in which to demonstrate their work.

From a thesis standpoint, this project has been very demanding on the knowledge I have learned over the past four years. This project has required me to learn several new languages (HTML, Perl, CGI) in order to build the programs. Because this is a continuation of a past thesis, it has made me use software engineering techniques to quickly learn the limitations and strengths of the existing code, and where modifications to the code fit in. This part alone is one of the fundamental tools a computer science engineer needs to know before entering a career. Finally, this thesis has also shown how much outside work must go into planning a thesis. This includes

meetings with the involved parties and promoting the thesis so as many people know about the system and will voluntarily use it as possible.

7-2 Findings

By submission time, the thesis will include four main points of interest: the new submission form that includes verification checks and a more usable interface; a Perl script used to catalogue the information into a MARC record; the transfer of all relevant files to the library server for permanent storage; and heavy promotion of the project to the fourth year engineering students. The programs will be working so that they are satisfactory to both the library and my technical advisor. The impact of the promotion will be determined by the percentage of students who use the system. Ideally, a ninety percent usage rate is an attainable target for this project (the other ten percent will have unforeseen technical difficulties or will choose not to use the system). Promotion will play a big part in increasing the use of the program from the thirty-three percent who used it in the fall TCC class.

7-3 Limitations and Future Research

Due to time constraints and resource problems, there were some limitations in the thesis. I was hoping to use registrar information to group the students into their respective TCC classes in order to make submission easier. Because of privacy laws, I was unable to gain access to this specific information. Instead, the thesis contains a scaled down version that includes only the student's name and identification number. Because the files must be portable to different servers, most of which can only understand HTML and Perl, the code is restricted to these languages. This project might have been easier to do if were done in another language designed for databases and user interaction, such as Visual Basic or Visual C++. Due to time constraints, some of the proposed additional features of the system had to be left out. These included allowing the TCC advisor to directly approve the thesis after it has been submitted so that there would not be any

written communication between the library and each TCC advisor. Another feature discussed but not implemented was designing a program that would analyze two documents and see if one had plagiarized the other.

There are many areas of future research involved with this thesis. The limitations described above, such as a document analyzer and having each TCC advisor submit the thesis, can be implemented. Additionally, work can be done to evolve the program into a school-wide thesis submittal form. There has been some interest to use the system to submit theses from departments outside of the engineering school. Allowing this will bring new challenges because of the added dimensions and options a program would need to service such a diverse user base. The availability of the software to more students could also improve. Increasing the number of licenses of Adobe's PDFWriter would allow students more computers to create the needed PDF files. Finally, there also needs to be more discussion and preparation between the library and the TCC department to make sure both parties know what is expected and to make sure electronic submission becomes a mandatory part of the thesis project, not a voluntary one.

Appendix A – Instructions

Instructions for Electronic Submission.

Note: Electronic Submission requires the use of Acrobat PDFWriter to convert the thesis document into a .pdf file. Ten computers in the Thornton Stacks have this software: computer numbers 001-005 and 009-013. If you do not have your own access to this software, please use these computers. Please fill out a Survey when you complete the submission so we can get some feedback from you on the whole process.

1. Open your completed thesis in your word processor.

2. Under "File", choose the "Print" command. Under "Name" or "Current Printer", click the down arrow. There will be a list of printers to choose from. Choose the second option from the bottom entitled "Acrobat PDFWriter." Then Choose "OK" or "Print."

3. A box will pop up requesting where you want to store the .pdf file. Put it into the same directory as the word processor file (usually, this is the default). Choose "OK" when finished.

4. Open Netscape Navigator. In the location bar, enter this URL: http://viva.lib.virginia.edu/etd/cgi-bin/etd.html. This will take you to the home page of the electronic submission form.

5. Follow the on-screen instructions, entering in the required information. Cut and paste any needed text from your **original** file for the title and abstract (The text in the abstract will be all on one line. It will correct itself later on so do not try and edit it.). Choose the full, actual name for yourself and your advisors. Click on the Preview button when finished. If you inputted an incorrect ID Number or you left out one of the fields, an error message will appear. Click on the "Back" button and correct the error.

6. Review the entries for each field. To correct any errors, press the Back button, make the changes, then press Preview again. If everything is correct, enter in the path of the .pdf file. Use the Browse function to locate it. A box will pop up for you to locate your .pdf file. Go to the correct directory where your file is, then under file name, type *.pdf. Your .pdf file will

appear. Choose it and press the “OK” button. When finished, click the “Send to Library” button.

7. You are done. The library will send an e-mail to you when your thesis is posted in VIRGO. If you wish to clear the information you entered, go back to the form page and hit the reset button.

Appendix B – Survey

I. Background Information

1. What type of computer did you use?
    a. PC
    b. Mac
    c. Other (Please Specify) -

2. What type of Word Processor did you use?
    a. Microsoft Word
    b. Wordperfect
    c. Framemaker
    d. Other (Please Specify) -

3. Did you have any special graphics or diagrams in your thesis (such as Visio, Excel graphics, wiring diagrams, test results, etc)? If so, please list these programs below.

II. The Submission Process

1. Were you able to submit your thesis without any problems?

2. If you answered "No" to #1, please list any problems that you had as precisely as possible. Please indicate which problems had no help associated with it and which had help that was confusing.

3. Was the process explained to you clearly, i.e. were the directions thorough and complete?

4. What part(s) of the process were confusing?

5. What part(s) of the process took the most amount of time?

6. What improvements can you suggest for making the submittal process easier?

7. Would you go through this process again if you had to?

Appendix C – Explanation of code

Because the bulk of the work for this thesis is modifying code in a set of programs, it is important to discuss some of the technical aspects of this project. This appendix is broken up into three sections, each a different script that I modified. Each section will discuss what the changes are actually doing as well as an explanation of the logic behind the code. A final section discusses random topics that came up while working on the project.

A-1 Changes to "etd.cgi"

The program "etd.cgi" is a Perl script that generates the html for the main submission page. It contains all the CGI boxes the student uses to enter in all their biographical information. There were three main modifications made to the program: the replacement of some text boxes with scroll boxes, the addition of an ID Number text box, and the removal of unnecessary code.

A-1-1 Creation of scroll boxes

The first change made was to eliminate the text boxes used for the student name and technical advisor, and replace them with scroll boxes. In addition, a scroll box was created for the TCC advisor. A text box is created using the HTML command:

```
$query->textfield(-name=>'name', -size=>'35')
```

This command simply creates a text field whose variable name is "name" and has a character size of thirty-five. To minimize the potential for incorrect name entry, these boxes have been replaced for the Student Name and Technical Advisor, and created for the TCC Advisor. Using the Student Name as an example, the following code produces a scroll box with the students' names:

```
$file = "/home/zdc3s/public_html/cgi-bin/student.sorted";
(1)
open(STUD_LIST, $file);                                    (2)
while ($name = <STUD_LIST>){                               (3)
     $name = substr($name, 10, length($name));
(4)
     @name_list = (@name_list, $name);                     (5)
}
close(STUD_LIST);                                          (6)

print $query->scrolling_list(-name=>'Student_Name',
(7)
                              -values=>[@name_list],
(8)
                           -size=>10);                     (9)
```

Line one assigns `$file` the string that is the directory where the student file exists. Line two actually opens the file and allows `STUD_LIST` to access it. The next set of lines (lines three through six) is a while-loop. While the loop condition holds true (`$name = <STUD_LIST>`), the body of the loop will execute. The condition loop does two things: (1) it tests to see if there are any values left in the file (2) and assigns the next value to the variable `$name`. Line four then reassigns `$name` to equal only the name contained in `$name`. For example, at line four, `$name` might contain the string "`123456789 Connelly, Zachary D.`" The scroll box only needs the names of the students, not their ID Numbers. Line four produces the necessary sub-string of `$name` by taking all the characters, starting at position ten in the string, through the end of the string.

After the name of the student has been created, line five assigns the value to the end of the array `@name_list`. It is inherent in Perl that an assignment like the one used adds the single value (`$name`) to the end of the array value (`@name_list`). The while-loop continues to add names to the array until there are none left in the file. Line six closes the file opened by line two. Finally, lines seven through nine constitute the CGI command that creates the scroll box. It creates a

scrolling list called "`Student_Name`" whose values are the members in the array `@name_list`. Line nine specifies how many names will appear in the box at any one time.

This technique was done for all three textboxes, only varying the name of the file and the name of the scroll box.

A-1-2 Creation of ID Number text box

The creation of the ID textbox is equivalent to the text field command discussed in A-1-1. The following code creates the ID Number textbox:

```
print "<b>ID Number:</b>",
(1)
     "\n<br>\n",
     "Enter your ID Number with no dashes.<b>  Use the form
     123456789</b>",
     "\n<BR>\n";

print $query->password_field(-name=>'ID_Number',
(2)
                                -size=>9,
                           -maxlength=>9);
```

The first print statement creates the static html used to guide the user to enter the ID Number. The second print statement is the actual CGI command. It is a password field instead of a text field because the ID Number needs to be protected from unauthorized viewing; when the student enters in the ID Number, the numbers are represented as asterisks. It has a character size of nine and the user can only enter nine characters into the box because of the `maxlength` field.

A-1-3 Removal of Code

In the old form, there was a text field asking the user how many files he or she wished to submit. The library and Professor French decided that this field is not necessary. Because students are able to include images and other files into word processor documents, the entire thesis can be one, all-inclusive file. Because of this, it is unnecessary to ask the user how many files are being submitted. Therefore, the code for this text field was deleted.

A-2 Changes to "etd0.pl"

The Perl script "etd0.pl" is the program that is called after the user presses the submit button on the submission form. This program receives all the information the user enters in the text fields and scroll boxes. Previously, this script did three things: check that all the required fields have information, check that the student said "Yes" to the question that the thesis is approved by both advisors, and then print all the information if the user passes both checks. The script prints an error message and exits if either of the checks fails.

To go along with the new form, some modifications were made. Two new features have been added: the script now compares the ID Number on record with the library against the one entered by the student and outputs the TCC advisor if all the checks pass. The latter feature was very easy to implement. I followed the same convention Miss Moore used to output information. The code looks like this:

```
print "<b>TCC Advisor:</b>",
      "\n<br>\n",
      $query->param('TCC_Advisor'),                    (1)
      $query->hidden(-name=>'TCC_advisor'),
      "\n<p>\n";
```

The important line in this code is line one. It inserts the name stored in the field `TCC_Advisor`. The rest of the code produces the static label surrounding the name.

The other major feature is the code that checks the ID Number entered by the student against the one on record with the library. The code looks like this:

```
$file = "/home/zdc3s/public_html/cgi-bin/student.sorted";
     (1)
open(STUD_LIST, $file);
     (2)
$flag = 'T';                                           (3)
while (($name = <STUD_LIST>) && ($flag eq 'T')) {
     (4)
    $stu_name = substr($name, 10, length($name));
     (5)
```

```
          if ($stu_name eq  $query->param('Student_Name')) {
           (6)
           $ID = substr($name, 0, 9);
                (7)
           $flag = 'F';                                          (8)
          }
     }
     close(STUD_LIST);
          (9)

     if ($query->param('ID_Number') != $ID) {
          (10)
     print "<H3>The ID Number you entered does not match that on
     (11)
           record.<P> <H1><B>You will not be able to continue

           until a correct ID_Number is entered.  </B><P>Press

           \"Back\" on your browser and re-enter the correct ID
           Number.</H1></H3>";

     print "\n</BODY></HTML>\n";
     exit;
     }
```

Lines one through eight find the ID number that corresponds with the student name. After opening the student file, there is a while-loop with the condition that (1) there are more names in the file and (2) the variable `$flag` equals “T”. `$flag` is a boolean variable, which is a variable that has only two values, true (T) or false (F). After entering the loop, line five produces just the student name and stores it into `$stu_name`. This name is then compared, using the if-statement, against the student name chosen by the user. If they are the same, lines seven and eight are executed. The ID number associated with student name is taken (using the same substring function as in A-1-1) and stored in `$ID`. The value of `$flag` is set to “F” so the while-loop will end immediately. Line nine closes the file.

The next sequence of lines checks the ID number taken from the file and compares it against the number entered by the student. This is done using the if-statement in line ten. If the if-statement is true (the numbers are not equal), the body of the if-statement is executed. This

body prints an error message alerting the user about the incorrect ID Number and instructs the student to go back. It will then exit the script. If the if condition is false (the two numbers are equal), normal execution of the program continues.

Looking back on this code, it seems like there is a way to do this task in far less lines and using a more efficient algorithm. Since I have very little experience in Perl, my goal was to produce the correct result as easily and logically as I could. For example, the boolean flag, is not really needed to end the while condition. Theoretically, there should be only one name that is chosen and the if-statement body should be entered only once. I felt the boolean statement was necessary so that the while-loop ends as soon as the match is made. If the student file is very large and the student chosen is at the top of the list, than many unnecessary comparisons have to be made after the correct name is found.

A-3 Changes to "etd2.pl"

The script "etd2.pl" is the program called once the librarian approves the thesis. It takes the information entered by the librarian, packages it with the biographical information and the thesis document, and sends it to its permanent storage site. The only change I made to this file was the code that makes the MARC record using the biographical information. Using the sample MARC record obtained from the library (Appendix E), I could see which parts of the record were constant and which were dependent on the student's biographical information. This code looks like:

```
open (MARC, ">>gen_record.txt");
     (1)

print MARC "*** DOCUMENT BOUNDARY ***\n";
@fields = spilt(/,/,$query->param('Date'));
     (2)
$Year = $fields[1];                                    (3)
$Year = substr($Year, 1, length($Year));
     (4)
```

```
print MARC "FORM=MRDF\n";
print MARC ".000. |",
    $query->param('Email'),
    "\n";
print MARC ".008. |a000000s",
     $Year,
    "    xx         d  000 0 eng d\n";

print MARC ".099.  |a";
if ($query->param('Dept') eq "Aerospace Engineering") {
     (5)
    print MARC "AE";
} elsif ($query->param('Dept') eq "Applied Mathematics") {
    print MARC "AM";
} elsif ($query->param('Dept') eq "Chemical Engineering") {
    print MARC "CE";
} elsif ($query->param('Dept') eq "Civil Engineering") {
    print MARC "CI";
} elsif ($query->param('Dept') eq "Computer Science") {
    print MARC "CS";
} elsif ($query->param('Dept') eq "Electrical Engineering")
{
    print MARC "EE";
} elsif ($query->param('Dept') eq "Engineering Science") {
    print MARC "ES";
} elsif ($query->param('Dept') eq "Mechanical Engineering")
{
    print MARC "ME";
} elsif ($query->param('Dept') eq "Systems Engineering") {
    print MARC "SE";
}
print MARC "|a",
#Catalogue Number
     (6)
      "|a(Online)\n";
print MARC ".100.  |a",
(7)
    $query->param('Student_Name'),
    print MARC "\n";
print MARC ".245.  |a",
(8)
    $query->param('Title'),
    "|h[computer file]/|c",
print MARC ".256.  |aComputer data (ca. ",
#File size goes here
     (9)
    "\n";
```

```
     print MARC ".260.  |c",
          $Year,
         "\n";
     print MARC ".516.  |aText and images (PDF)\n";
     print MARC ".538.  |aMode of access: Internet.\n";
     (10)
     print MARC ".500.  |a",
         $query->param('Dept'),
         "-TCC402\n";
     print MARC ".502.  |aThesis (B.S.)--University of Virginia,
",
          $Year,
         ".\n";
     print MARC ".590.  |aAdvisors: ",
         $query->param('Technical_Advisor'),
         " and ",
         $query->param('TCC_Advisor'),
         ".\n";
     print MARC ".856 42|zAbstract only\n",
         "|u",
     #Hyperlink location
         "\n";
     print MARC ".856 40|zFull text, Acrobat Reader required\n",
         "|u",
     #Hyperlink location
         "\n";
     close (MARC);
```

This script appends to the end of a file called "`gen_recod.txt`" a new record. Most of the lines are print commands that write lines to the file. These lines are made up of constant texts that will never change (such as line ten) and information specific to each record (such as the title and author, lines seven and eight respectively).

Two sections in the code are of interest. The first section is lines two through four. These create the year that the thesis was submitted. Because the theses might not be catalogued in the same calendar year as they were submitted, it is important to use the submission date entered by the student. However, the library only requires the year, not the month or day. Lines two through four extract away the unnecessary parts of the date, producing the year. Line two splits the full date into two pieces – all the characters before the comma and all the characters after the comma –

and puts these new strings into the array `@fields`. Line three assigns the second string in `@fields` to the variable `$Year`. Because the year is preceded by a space (March 24, 1998), it must be removed from the string. Line four does this by taking the sub-string from the first character position (the first digit) to the last character in the string (the fourth digit) and stores it back into `$Year`.

The second modification is in the if-elsif-statement block located at line five. The MARC record needs to know which department the author is, and represent it using a two-character abbreviation. The if-elsif block uses the student's department and checks it against all the possible departments. When there is a match, the corresponding two-character symbol is added to the line.

A-4 Additional notes on programming

In addition to the changes made to the files above, there are also a few "random" ideas that occurred while programming. The first is the distinction between dynamic and static HTML. When the "etd.cgi" script is run, it is very slow because of the large volume of names it must process for the scroll boxes. Because the system should be as usable as possible, a lengthy delay will hurt the chances of the system being used by students. The script is in Perl, but it creates an html page just like any other. This html page is static, meaning that if the names of the teachers changed, the constant html page would not change as well. The html page will change only when "etd.cgi" is executed first. However, the html page runs quicker because it is constant. Therefore, when the system is finished and all the changes are made, "etd.cgi" will be executed first. The html page it creates must be saved and this page will be the submission form for the system.

The second consideration taken is associated with the previous idea. All the scroll boxes are based on text files. This allows the boxes to be updated whenever necessary – probably every semester to accommodate the new students who would use the system. The library will only need

to copy the text files into the appropriate directories and run the “etd.cgi” script once to produce the new html submission form.

Appendix D – Screen shots

As stated in Chapter 5, this form can be reached at:

http://viva.lib.virginia.edu/etd/cgi-bin/etd.html

D-1 Submission Form

University of Virginia SEAS Thesis Submission Form

Instructions: Please fill out the form completely. Cut and paste, from your document and into the form, as necessary. Read the help file for help on cutting and pasting your abstract. Once you are done filling out the form read the copyright statement at the bottom of the page and if you agree to it click "Preview".

Name:

Choose your name from the scroll box below.

CONNELLY, ZACHARY D.
DEASON, KRISTIN S.
GIBSON, JASON T.
PRESTWOOD, GARRET C.
SHAMAS, JAMES A.
VANANTWERP, COLIN F.

ID Number:

Enter your ID Number with no dashes. **Use the form 123456789**

E-mail address:

Enter your full e-mail address. **Use the form abc1d@virginia.edu**

Adjust the width of your browser so that both ends of the "Title" field are visible.

Title:

Enter the title just as it appears on the title page.

Department:

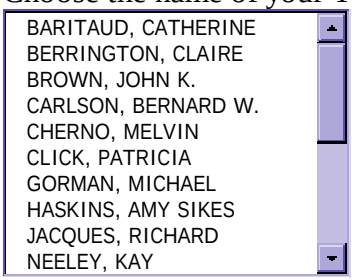
- (•) Aerospace Engineering
- ( ) Applied Mathematics
- ( ) Chemical Engineering
- ( ) Civil Engineering

- Computer Science
- Electrical Engineering
- Engineering Science
- Mechanical Engineering
- Systems Engineering

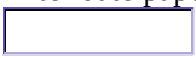
Technical Advisor:
Choose the name of your technical advisor

A scroll box should be here. Word can not put the entire scroll box in because it is so big.

TCC Advisor:
Choose the name of your TCC advisor



Date of approval:
Enter date paper was approved by your advisors. **Use the form September 10, 1995**



Abstract:
Enter your abstract below. If you are cutting and pasting **be sure to read the help file section on abstracts first!** If you do not it is likely that your abstract will look wrong when publicly displayed on the WWW.

Copyright Statement:
I hereby grant to University of Virginia or its agents the right to archive and display my thesis or

dissertation in whole or in part in the University Libraries in all forms of media, now or hereafter known. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

◉ I agree

Signed Title Page:
I realize that my submission will not be processed until I give the TCC Department a **printed** copy of my title page that has been **signed** by both my TCC and Technical advisors.

○ Yes ◉ No

Preview:

Press "Preview" to continue. [Preview]

Press "Reset" to erase all the values. [Reset]

D-2 Student approval form

UNIVERSITY SEAS Thesis Submission Form - Preview

Instructions: If everything below is correct then send it to the library by clicking "Send to Library" below. If something needs to be changed press the "Back" button on the browser, change what needs to be changed and then press "Preview" again. Please save (*do not* "Bookmark") this page for your records.

Name:
CONNELLY, ZACHARY D.

Email address:
zdc3s@virginia.edu

Title:
Test page used for the thesis

Department:
Computer Science

Technical Advisor:
FRENCH, JAMES

TCC Advisor:
RUSSELL III, EDMUND P.

Date of approval:
March 24, 1998

Abstract:
This is being used for the appendix to my thesis to show what this page looks like.

Select the file to be sent using the "Browse" button. If you do not see a "Browse" button, click here.

If everything above is correct click "Send to Library".

Send to Library

D-3 Error message if ID Number is incorrect

The ID Number you entered does not match that on record.

You will not be able to continue until a correct ID_Number is entered.

Press "Back" on your browser and re-enter the correct ID Number.

Appendix E – Example MARC record

FORM=MRDF
.000. |aam9a
.008. |a000000s1997 xx d 000 0 eng d
.099. |aCS|a970008|a(Online)
.100. |aBodayla, Roy J.
.245. |aUsing software reuse to develop a Z specification tool
|h[computer file] /|cRoy J. Bodayla.
.256. |aComputer data (ca. 127 kilobytes)
.260. |c1997.
.516. |aText and images (PDF)
.538. |aMode of access: Internet.
.500. |aComputer Science-TCC402.
.502. |aThesis (B.S.)--University of Virginia, 1997.
.590. |aAdvisors: John C. Knight and Melvin Cherno.
.856. 7 |zAbstract only
|uhttp://univac.cs.virginia.edu:3066/Dienst/UI/2.0
/Describe/uva.seas.cs%2fCS97-0006|2http
.856. 7 |zFull text, Acrobat Reader required
|uhttp://univac.cs.virginia.edu:3066/Dienst/Reposi
tory/2.0/Body/uva.seas.cs%2fCS97-0006/pdf|2http

Bibliography

Benyon, David and Phillipe Palangue. 1996. Critical Issues in User Interface Systems Engineering. New York: Springer-Verlag Limited.

--This book discussed different areas in interfaces that programmers don't think about to test for errors. Users will try to do something in the interface that was not thought about, and the whole system crashes.

Bienz, Tim and Richard Cohn 1993. Portable Document Format Reference Manual. New York: Addison-Wesley Publishing Company.

--A manual by Adobe Systems describing PDF. This is the file format used by the system to submit the electronic theses.

Davis, James R. and Lagoze, Carl 1994 A protocol and server for a distributed digital technical report library. On-line. Internet. http://cstr.cs.cornell.edu/Dienst/UI/2.0/Describe/ncstrl.cornell%2fTR94-1418?abstract=

--On-line documentation for the Dienst protocol. Includes the rationale for the project, some examples on how to use it, and an explanation for some of the decisions the design team made.

"Dienst Installation/Maintenance Documents." (November 1996). On-line. Internet. http://ncstrl.org/Dienst/htdocs/admin_doc_menu.html.

--This document is a follow-up version of the initial protocol detailing additions and any problems in the first version that were solved.

"ETD: Electronic Thesis and Dissertation Process." February 22, 1997. On-line. Internet. http://etd.vt.edu/etd/.

--On-line documentation on Virginia Tech's on-line thesis system. It includes the rationale for the project, the technical specifications, and other information used in the development for the University of Virginia's system.

Fowler, Susan L. and Victor R. Stanwick. 1995. The GUI Style Guide. Boston: AP Professional.

--This a general purpose book detailing some of the basics in GUI design.

Fox, Edward A., John L. Eaton, Gail McMillan. September 1996. "Improving Graduate Education with a National Digital Library of Theses and Dissertations." A Proposal Submitted to the United States Department of Education.

--This is a government proposal outlining the need for a digital library to hold national theses and dissertations. It is a good source for some of the fundamental rationale of this project.

Fox, Edward A., John L. Eaton, Gail McMillan. September 1996. National Digital Library of Theses and Dissertations: A Scalable and Sustainable Approach to Unlock University Resources. On-Line. D-Lib Magazine. http://www.dlib.org/dlib/september96/theses/09fox.html.

--This is an on-line journal article discussing the background and rationale for national thesis digital library. It discusses some history of digital libraries, the rationale and impact of a national library system, as well as some of Virginia Tech's future plans.

Fox, Edward A., John L. Eaton, Gail McMillan. Septmber 1997. Networked Digital Library of These and Dissertations: An International Effort Unlocking University Resources. On-Line. D-Lib Magazine. http://www.dlib.org/dlib/september97/theses/09fox.html.

--This is an on-line journal updating the past work at Virginia Tech on the NDLTD. It includes the progress made so far, some debate on controversies that have arisen, as well as the future growth of the program.

Galitz, Wilbert O. 1993. User-Interface Screen Design. Boston: QED Publishing Group.

--This book provided some of the basics in interface design.

Harter, Stephen P. 1997. Scholarly Communication and the Digital Library: Problems and Issues. Journal of the American Society for Information Science

--A discussion on exactly what a "digital" library is. Includes the origins of digital libraries, what the primary properties of a digital library are, and what some of the problems facing digital libraries are.

Hix, Deborah and H. Rex Harrison. 1993. Developing User Interfaces: Ensuring Usability Through Product and Process. New York: John Wiley & Sons, Inc.

--A book on what usability is and how it affects user interfaces. Provides theories and methods to make user interfaces simple and natural.

Mikasa, Francis 1996. The Cultural Legacy of the "Modern Library" for the Future. Journal of Education for Library and Information Science 37: 100-119.

--Discussion on the definition on what digital libraries are. This author thought that digital libraries should also be personal information collections.

Moore, Mariahna 1997. "Electronic Submission of Undergraduate Theses: A Case Study for the Implementation of a University-Wide Program for the Electronic Submission and On-Line Archiving of Theses and Dissertations." Diss. University of Virginia.

--This is a UVA student's thesis from last year. She developed the prototype version of UVA's ETD. It includes some of her design issues and decisions she went through in developing the system.

"Networked Computer Science Technical Reports Library." October 1996. On-line. Internet. http://www.ncstrl.org.

--This is another example of digital libraries already in place. It describes the national system of dissertations that are held on-line at universities across the country.

Schwartz, Randal L. 1993. Learning Perl. Cambridge: O'Reilly & Associates, Inc.

-- This book describes the Perl Programming language. It includes the basic conventions of the language as well as some of the more advanced topics in the language.

Seaman, David. July 1997. The User Community as Responsibility and Resource; Building a Sustainable Digital Library. On-Line. D-Lib Magazine. http://www.dlib.org/dlib/july97/07seaman.html.

--This on-line journal article discusses an example of on-line archives. It is based on the Electronic Text Center at the University of Virginia.

Weber, Thomas E. and Robin Frost. September 26, 1996. "Watching the Web." The Wall Street Journal. B6.

--This article gave some statistics about Internet users and the growth of the Internet over the last three years.