



Theses and Dissertations

2009-10-12

Research in Information Technology: Analysis of Existing Graduate Research

Christopher John Cole
Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Engineering Commons](#), and the [Library and Information Science Commons](#)

BYU ScholarsArchive Citation

Cole, Christopher John, "Research in Information Technology: Analysis of Existing Graduate Research" (2009). *Theses and Dissertations*. 1935.
<https://scholarsarchive.byu.edu/etd/1935>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Research in Information Technology: Analysis of Existing
Graduate Research

Christopher John Cole

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Joseph J. Ekstrom, Chair
C. Richard G. Helps
Barry M. Lunt

School of Technology
Brigham Young University
December 2009

Copyright © 2009 Christopher John Cole

All Rights Reserved

ABSTRACT

Research in Information Technology: Analysis of Existing
Graduate Research

Christopher John Cole

School of Technology

Master of Science

Information Technology is an academic discipline that is well recognized by the academic community. There is an increasing number of schools offering degrees in Information Technology and has there is an official curriculum published with the ACM/IEEE computing Curriculum. A concern with Information Technology as an academic discipline is that it does not have a clearly defined set of research issues which are not studied by any other discipline. One way to propose this set of issues is to perform a "bottom-up" analysis and gather research in IT that has already been published. This research can then be analyzed for recurring themes.

This research describes a repository of graduate level work in the form of master's degree theses and projects and doctoral dissertations. A keyword analyses was done on the publications gathered, and it was confirmed that a set of themes could be found. As a demonstration of the viability of this approach the methodology has identified five initial themes. A larger sample is required to define a definitive set of themes for the IT discipline.

Keywords: Information Technology, graduate research, repository, keyword analysis, research agenda

ACKNOWLEDGMENTS

I would like to thank the combined efforts of every member of the IT program. Their doors were always open for me, and I was willingly given help, advice, and encouragement. My graduate committee in particular was invaluable and readily made themselves available when I needed them. My committee chair, Dr. Ekstrom, is who made everything possible. The countless hours given and the personal interest shown for this research are a testament to his dedication to his students.

My family also deserves a great thank you for their support at every step. I have always been able to count on them. Thank you so much for your help and inspiration.

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	xi
1 Introduction.....	1
1.2 Problem statement.....	3
1.3 Hypothesis	4
1.4 Justification.....	4
1.5 Assumptions.....	4
1.6 Delimitations.....	5
2 Review of Literature	7
2.1 Identity crisis in other disciplines	7
2.2 Issues to be recognized as an academic discipline	8
2.2.1 Bottom-up approach.....	8
2.2.2 Top-down approach	9
2.2.3 Issues to be resolved	10
2.3 Keyword analysis.....	13
3 Methodology	15
3.1 Determine a representative sample	16
3.2 Create the repository	17
3.3 Test the repository	18
3.4 Gather data and populate the repository	18
3.5 Analyze the results and test the hypothesis	19
4 Results and Data Analysis	21
4.1 Gathering data.....	22

4.1.1	Contacting schools	22
4.1.2	Determine a representative sample	23
4.1.3	Data from individual institutions	26
4.2	Google app engine	28
4.3	User testing of the repository	29
4.4	Keyword analysis.....	30
4.4.1	Thesis keywords from titles	32
4.4.2	Thesis keywords from keywords	36
4.4.3	Thesis keywords from abstracts	39
4.4.4	Thesis keywords from everything.....	42
4.4.5	Thesis keywords – conclusions.....	43
4.4.6	Citations of theses	46
4.5	Other observations	46
4.6	Direct intuitive analysis	47
4.7	Proposed themes in IT	48
4.8	Comparison to other proposed themes.....	49
4.9	Comparison to 5 pillars of IT.....	50
5	Conclusions and Recommendations	51
5.1	Overview.....	52
5.2	Interpretation of findings	52
5.3	Recommendations for action	53
5.4	Recommendations for further study	54
5.5	Closing	54
	References.....	55
	Appendix A. Graduate Publication Keyword Lists.....	57

Appendix B. Citations Keyword Lists	73
Appendix C. Google App Engine Code.....	81
Appendix D. Browse and Search Page Code.....	143

LIST OF TABLES

Table 4-1: Schools contacted	22
Table 4-2: Universities with responses	23
Table 4-3: Thesis distribution by institution	28
Table 4-4: Title keywords from titles	31
Table 4-5: Thesis keywords from keywords.....	35
Table 4-6: Thesis keywords from abstracts	37
Table 4-7: Thesis keywords from everything	39
Table 4-8: Citations keywords from everything	44

LIST OF FIGURES

Figure 2-1: CC2005 Graphic Showing IT Equal with Others	11
Figure 4-1: Pillars of Information Technology	50

1 INTRODUCTION

1.1 Background

Information Technology is a relatively new discipline, with its research areas and educational boundaries still evolving. While there are some who are hard at work trying to continue the development of the academic discipline of Information and its definitions (Helps and Ekstrom 2008; Lunt and Ekstrom 2008), others are still skeptical. There are various other areas of research in computing, such as Computer Science, Information Systems, Computer Engineering, etc. Each successful discipline carves out its own set of topics and definitions and then evolves them over time.

There has been work done (Aldrich 1999; Benbasat and Zmud 2003) to suggest that in order for an academic discipline to be established, there are two issues that must be resolved: the learning issue and the legitimacy issue. The learning issue tests whether a discipline can identify a coherent set of issues and develop its own set of methodologies for approaching them. The legitimacy issue requires that the discipline be recognized as a separate discipline from the environment in which it operates.

The legitimacy issue has been successfully resolved. Reichgelt (Reichgelt 2004) argues that an increasing number of schools are establishing undergraduate programs in Information Technology in addition to other computer disciplines. Presumably, this has been done because of a demand by potential students and administrators, showing recognition for IT in the academic world, and to some extent in the commercial world, since those graduates are finding

professional level employment. Ekstrom (Ekstrom et al. 2006) argues that IT is now recognized as a separate discipline by the Computing Accreditation Commission of ABET, putting it on par with Computer Science and Information Systems. Finally, the IT community has published the IT curriculum (Lunt et al. 2008), which is part of the ACM/IEEE Computing Curriculum (Shackelford et al. 2006).

The discipline of IT has not yet been as successful at resolving the learning issue. The simplest way to resolve this issue is to identify a set of phenomena that are not studied by any other discipline (Reichgelt 2004). The discipline of Information Technology has been unsuccessful in this regard thus far.

There are two ways of resolving this issue (Ekstrom et al. 2006). One way is to take a “top-down” approach by defining a research agenda based on a theoretical analysis of the discipline. The other way to resolve the issue is through a “bottom-up” approach. This method defines a research agenda by looking at the peer-reviewed research that has been published by the community and drawing conclusions. The problem with this is that, as of yet, there are few peer-reviewed journals devoted to IT research. Most research that happens in IT gets published in other journals, mixed in with other academic disciplines. At first look, it appears as though a bottom-up approach will not be possible for defining a research agenda in IT.

Ekstrom proposed a solution to this issue (Ekstrom et al. 2006). There is in fact a body of research that is reviewed and accepted by a committee of experts in the field: master’s theses and dissertations from reputable graduate schools. He has gathered a set of 70 master’s theses from Brigham Young University and Purdue University and given each one a classification based on the title and abstract. One goal of his analysis was to find evidence of an implicit IT research

agenda. The evidence shows that, in at least these two schools, there is an implicit IT research agenda, and more work in the area would in fact be useful.

The styles used in this template are outlined in the next several sections.

1.2 Problem statement

Information Technology, in order to establish itself as an academic discipline, must resolve the learning issue. One way to resolve this issue is to take a bottom-up approach by studying peer-reviewed papers published in the field of IT, looking for recurring themes and publishing a research agenda from the data presented. Master's theses and dissertations from reputable graduate schools provide an excellent venue for such a study of peer-reviewed research papers.

Ekstrom's research (Ekstrom et al. 2006) provides preliminary evidence suggesting that there is an implicit research agenda in Information Technology. This initial analysis of 70 theses from two schools suggests that research on a larger scale would be useful. Such a research project requires the creation of a repository that is easily accessible to all those willing to participate in this research, and also requires the proposal of appropriate analysis approaches to identify a possible research agenda.

Once such a repository is created, a more in-depth analysis of the greater IT discipline can be done. Such an analysis will be able to validate and improve upon the research that has already been completed.

1.3 Hypothesis

Given a repository of IT research, a keyword analysis would suggest meaningful themes for an IT research agenda.

1.4 Justification

The academic discipline of Information Technology is evolutionary. Not only is it currently being developed and defined, but because of the nature of technology, it is also changing at the same time. The borders can be gray and fuzzy, but finding and developing a core research agenda will help IT in its early stages of development.

Creating a repository of IT research will also be of great use for several reasons beyond that of this particular research. First, it will continue beyond a single thesis project. It will provide an information source organized for research available to all interested parties. Potential employers can look through the research of a particular advisor, student, or of Information Technology as a whole in order to get a more clear idea of what to expect from an IT professional. Other researchers and students can use the repository for inspiration on new theses as well as old ones to build upon. “The ability to stand on the shoulders of those who have gone before is a sign of a more mature academic discipline.” (Ekstrom et al. 2006).

1.5 Assumptions

This research assumes that master’s theses and dissertations published by reputable graduate schools offer a good representation of research in the IT community as a whole. This research also assumes that the identified graduate publications are a good representative sample of the entire group of IT graduate publications.

1.6 Delimitations

The repository is going to be generally available and easily updated. This will allow for a much greater number of articles to be submitted, but it also relinquishes ultimate control to the IT research community. While care will be taken to ensure that all of the information in the repository is accurate, it is possible for errors to occur.

This repository will initially contain as many master's theses and dissertations from as many schools as possible using the methodology described. This research is not an exhaustive survey; however the establishment of this repository will allow the collection of research documents and their analysis to continue to grow indefinitely after the conclusion of this specific project.

2 REVIEW OF LITERATURE

Information Technology as an academic discipline has seen a steady increase in recognition for the past few years. There are many who are happy about this development and believe that IT should continue to emerge on its own as an academic discipline; there are also many who believe that IT adds nothing new. (Reichgelt 2004).

This has been an issue in academia for years. The argument against IT contends that it is only a weakened subset of another discipline such as Information Systems, Computer Science, or possibly even Computer Engineering. One claim is that the IT community has yet to clearly communicate its common goals and expertise, essentially failing to adequately describe why it is distinct from these other disciplines.

2.1 Identity crisis in other disciplines

The fact is some of these other fields are having some of their own identity crises (Alford et al. 2004; Benbasat and Zmud 2003). Computer Science has published curriculum guidelines in 1968, 1978, 1991, 2004, and have expressed intent to continue with yet another revision. They hope that another revision will help define the differences between Information Technology, Computer Science, Computer Engineering, Information Systems, and Software Engineering.

2.2 Issues to be recognized as an academic discipline

There is work by Aldrich and others (Aldrich 1999; Benbasat and Zmud 2003) that suggests that in order for a discipline to become a recognized entity on its own, the discipline must resolve two main issues. The first issue is called the “learning issue.” The learning issue tests whether a discipline can identify a coherent set of issues and develop its own set of methodologies for approaching them. The learning issue has two parts: the “core issue” and the “distinctiveness issue.” The core issue means there is a specific set of questions and research methodologies that can reasonably be claimed by a particular discipline. The distinctiveness issue is when these questions and methodologies are indeed distinct from those of other disciplines.

The second issue is the “legitimacy issue.” This requires that the discipline be recognized as a separate discipline from the environment in which it operates. There are two parts to the legitimacy issue: cognitive legitimacy and sociopolitical legitimacy. Cognitive legitimacy describes when others in the academic community recognize the discipline as a separate discipline with a separate set of issues that are not addressed by anyone else. Sociopolitical legitimacy is when the discipline is accepted by a few key groups of people. This includes the general public, governmental agencies, and academic administrators.

2.2.1 Bottom-up approach

Much research in the academic field of IT has been done with the importance of the learning issue and the legitimacy issue in mind. (Benbasat and Zmud 2003; Ekstrom et al. 2006; Lister and Box 2008; Miller et al. 2006). Some are using what is called a bottom-up approach to this research as opposed to a top-down approach. (Ekstrom et al. 2006; Reichgelt 2004). The

argument is that the discipline of IT already exists in many respects, and with the current IT research and in particular all of the IT programs across the country, there is a lot of data already available. So rather than a pure top-down approach where there is an agreed upon definition of a subject beforehand, a bottom-up approach would provide input and propel more formal definitional work forward. In a bottom-up approach, the currently available research is gathered and used to create more specific conclusions on what the IT discipline is really all about.

The distinctiveness issue has been written about using the bottom-up approach, researching the similarities and differences in baccalaureate programs in a set of 11 institutions (Reichgelt et al. 2004). This study compares the subjects studied in the programs of Information Technology, Information Systems, and Computer Science. The results give very clear definition as well as separation for these three programs. Information Systems shows a clear emphasis in Business along with some Networking, Web, & Databases and Software. Computer Science shows a clear emphasis in Software, Math and Science. Finally Information Technology has a definite emphasis in Networking, Web & Databases with some Software, Math, Science and Interpersonal Communication.

2.2.2 Top-down approach

There has also been significant work at defining IT from the top-down. In 2003, Lunt and Ekstrom printed a formal document describing what the core of an IT education should include (Ekstrom and Lunt 2003). They propose that IT is a seam between technologies. The “pillars” of an IT education that they propose are:

1. Programming
2. Database design, deployment and management.
3. Web systems, design, deployment and management.

4. Network design, deployment and management.
5. Human Computer Interfacing, including user advocacy.

The primary goals of IT are centered around the use, deployment and management of devices and services on the network. This is the big difference between Computer Science and Information Technology. For example, in IT programming is a means to an end while in CS, programming is a primary area of study. This is why IT programs are much more project and lab-oriented and much less theory based than CS programs. The official IT curriculum guide (Lunt et al. 2008) says that Information Technology “is concerned with issues related to advocating for users and meeting their needs within an organizational and societal context through the selection, creation, application, integration and administration of computing technologies.”

Work has since continued in this area, revising IT programs (Helps and Ekstrom 2008) and continuing to identify the IT professional (Denning 2001).

2.2.3 Issues to be resolved

Much of the legitimacy issue has already been resolved. Many universities across the country recognize IT as a separate discipline, offering degrees in IT in addition to other degrees such as Information Systems and Computer Science. (Reichgelt et al. 2004; Shackelford et al. 2006). Figure 2-1 shows how the Computing Curricula 2005 document puts Information Technology on equal footing with four other computing disciplines. The Computing Accreditation Commission of ABET recognizes IT as a separate discipline and has begun accreditation of the different programs (Ekstrom et al. 2006; Shackelford et al. 2006). A curriculum has been formulated and is being maintained and updated (Lunt et al. 2008). The

evidence is there to claim that IT is recognized as a separate discipline by the other disciplines, effectively resolving the legitimacy issue.

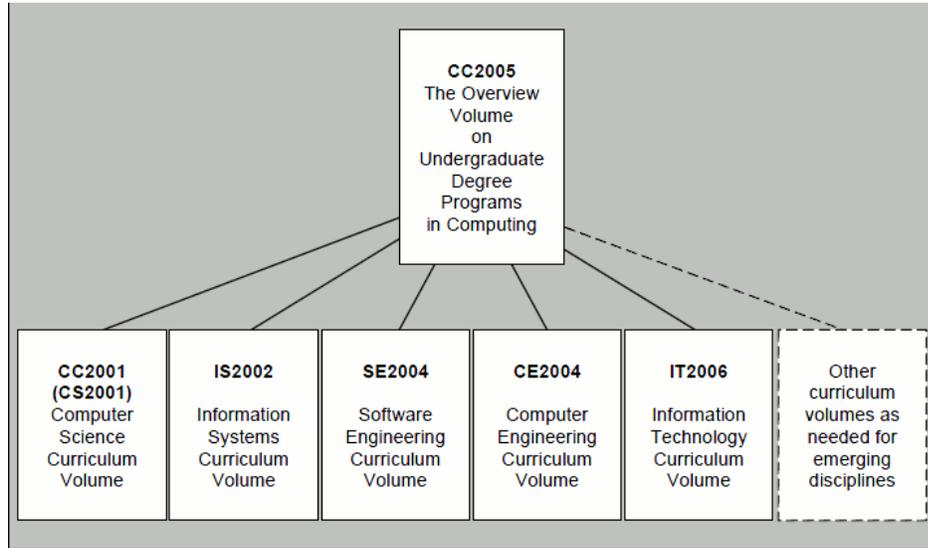


Figure 2-1: CC2005 Graphic Showing IT Equal with Others

IT has been less successful at resolving the learning issue, however. There is no formal research agenda for IT, defining a set of issues and methodologies unique to the discipline.

One suggestion is to use recurring concepts, defined as significant ideas, concerns, principles and processes that help to unify an academic discipline. (Alford et al. 2004). These recurring concepts should pervade the discipline and be independent of any particular technology. While as of yet there are not enough conferences or peer reviewed journals devoted solely to IT research to use as a basis to find these recurring concepts, another medium has been suggested (Ekstrom et al. 2006). Master’s theses and dissertations from reputable graduate schools are reviewed and accepted by experts in the field, giving a body of documents from which to draw conclusions on recurring concepts. This aligns with the bottom-up concept

already discussed, using the research done by graduate students to discover the recurring concepts. This same body of research links to other IT research through their references as well as related work done by the supervising professor of the graduate student.

The bottom-up approach of analyzing graduate theses and dissertations will be a good example of current IT research, including a specific set of issues and methodologies. A preliminary review gave promising results (Ekstrom et al. 2006), and the authors found that the graduate publications they analyzed fit into one or more of the following five “clusters”:

- DEV: Development projects (case studies)
- ED: Education and IT for Education Domain
- IAS: Information Assurance, Security, and Forensics
- PROJ: Project Management and IT for Project management domain
- TECH: Technology Evaluation and Modeling

They also were able to see the influence of particular professors on the research of their students. The cluster in which a thesis fit is often related to the research interests of the professors involved. The author feels that their preliminary review warrants a wider, more in depth research project, and that such research would be valuable to the academic community. That is what this thesis intends to provide.

There exists a repository of a similar nature to that called for by Ekstrom for the field of educational technology. Ed Cafarella of SUNY College at Cortland has compiled, and regularly updates, a “directory” of doctoral research dissertations (Cafarella). Graduate students and faculty refer to this well-known resource frequently as they create new research and monitor the scholarly work of colleagues and research initiatives. The author describes it as follows; “One

key element in the continued development of the field and profession of instructional design and technology is the research being done by graduate students.”

2.3 Keyword analysis

Keyword analysis is has been used by many researchers as a valid statistical analysis method. One study by Finn (Finn 1977) states that some of the most frequent words are “low information” words. For this reason, the low information words in a keyword analysis should be taken out. Finn also stated that, “rare words that are repeated in a text are closely associated with the topic of the text.” This is also backed up by other works (Carroll and Roeloffs 1969). Therefore a study of word frequencies in a body of IT documents should yield certain higher frequency words that dictate the topics of the publications analyzed (after the “low information” words have been taken out).

3 METHODOLOGY

The purpose of this study is to expand upon existing preliminary work (Ekstrom et al. 2006). The earlier study took a set of 70 master's theses from two schools, looking for patterns of research, suggesting an implicit research agenda within the academic discipline of Information Technology. The earlier research gave promising results, and a similar study on a larger scale was called for. In order to facilitate this kind of research, Ekstrom called for a repository to be made in which master's theses and dissertations could be easily added, searched through, sorted, and analyzed. The work that was to be done is as follows:

1. Determine a representative sample of theses for a valid study.
 - It was estimated that at least 200 theses from 5 different graduate schools would approximate a representative sample.
 - Contact 5 or more graduate schools to gather information on how many theses they publish each year in order to get an estimate at how many are published nationwide.
 - i. Each contact was asked to provide other graduate schools/contacts.
 - ii. Each contact was also asked to participate in the study by either entering data into the repository or making the data available for entry.
2. Create and test a repository for graduate level theses and dissertations.
 - The mechanism for gathering this data was a website.

3. Gather data and populate the repository.
 - Theses and dissertations from 5 or more academic programs were to be entered into the repository.
 - The website was to provide the ability to edit and correct information in the repository to make the information as complete and accurate as possible.
4. Analyze the results, test the hypothesis.
 - Generate keyword frequency analyses and propose themes.
 - Provide a proof of concept that a cluster analysis can be done (through a search function).

3.1 Determine a representative sample

The goal of creating a repository of IT scholarship is to provide access to the body of work in a single venue. Valid conclusions can only be drawn if the database contains a representative sample of the theses published. The total number of theses published was not known. Therefore, the problem was to determine what a representative sample would be.

The initial estimate was that at least 200 theses from 5 different graduate schools will approximate a representative sample. This was a rough starting figure coming from a rough understanding of the number of students in schools across the country. In order to come to a better grounded estimate, other schools with Information Technology programs were to be contacted. Using the resources available from the members of SIGITE and other active researchers in this field, a list of reputable graduate schools was to be made. These schools were also to be asked to provide information about additional graduate schools that should be included in the research. Each graduate school was to be asked to provide statistics on how many theses

they publish each year. Using this data, the initial estimate was to be improved and a representative sample determined.

3.2 Create the repository

This repository needed a platform with some specific characteristics in order for it to be useful. Some important characteristics include:

Ability to grow with use: The idea is that this repository will become increasingly well known and used in the research community of Information Technology. As it does, the database will continue to grow. With increased database size, the search functions and other features must scale to provide equivalent access.

Longevity: The repository is not meant to be temporary. The repository was to be created so that it would last more than 5 years without significant upkeep.

Stability: The repository needs to be available over the Internet at all times, with no significant downtime. Any downtime will discourage other researchers from using it, defeating the purpose of such a repository.

Database Modeling: The repository needed to have a design which will allow theses to be indexed and sorted according to a number of settings, including searching by title, keyword, abstract, etc. Also, it was to allow other groupings so that a researcher can see a timeline of one paper leading to another, clusters of research around schools, advisors, etc.

Inexpensive: The repository needs to be cost effective. This repository will be maintained by volunteer and graduate workers. Initially there is no budget to support the maintenance of the repository. As the repository grows and a university budget can take part in maintaining it, these costs should be minimal.

With all of these characteristics in mind, Google App Engine was chosen using Python as the backend programming language (Google). Google App Engine provides much of the necessary functionality. The website and data is spread across many different servers with load balancing and other important scalability features built in to ensure data uptime and an almost unlimited ability to grow. The Google App Engine API provides a simple interface to the backend, essentially taking care of all the work to keep the data organized in this manner. The application is running on the same backend as other Google applications, giving the same reputation for reliability and performance as Google. The database allows for objects to be stored into a database, allowing such things as inheritance in objects stored.

3.3 Test the repository

One key point of this repository is that it must be easy to use, and the people who desire access will be able to get started immediately without training. User testing was to be done, both with non-technical people as well as stakeholders in the area of IT research. They were to be given the task to add a thesis to the repository, with the results reviewed and any suggestions made to improve their experience.

3.4 Gather data and populate the repository

Once the repository was created and made accessible, the data input began. The repository was to be accessible to anyone on the Internet, but much of the work was and will have to be provided by the researchers.

The first step was to find out where the researcher needs to go to access the theses and dissertations existing at graduate institutions. Much of this work was to be done while

determining a representative sample. Any graduate school found with published theses available was to be included in the study. Instructions for accessing theses and dissertations at each of these schools were to be written. This makes it so that anyone who desires to participate will be able to do so. This also required gathering contact information from professors or administrators and getting a list of theses or instructions on how to find these theses from the professors. One good resource to start from was the ABET list of accredited schools in IT. All of these programs are undergraduate programs, but some also offer graduate degrees or could be in contact with other institutions that do.

The next step was to populate the database. This was to be done by the researchers and administrative personnel at the participating graduate schools. Those accessing the database were to have the ability to add new research as well as edit erroneous entries. As data was gathered and the database populated, the theses were to be sanitized and corrected as needed on the website. It was expected that there would be some human error in the database input, so there were to be features to review and correct those errors.

3.5 Analyze the results and test the hypothesis

Using searches on the title, keywords, and abstract, the clusters of research interest were to be created. The basis for this search was to begin on the five characteristics found in the preliminary research (Ekstrom et al. 2006), but these characteristics were revised. The following are some tests that were to be applied to the data:

- Word histogram on title, keywords, abstract: Create a list of the most used words in the title, keywords, and the abstract. After throwing out the common words that aren't helpful (e.g. the, this, a, for, etc.), this was to help to identify categories.

- Manually categorize each thesis: Once categories were chosen, each thesis was to be manually categorized, and then the automatic and manual categorizations would be compared.
- Cluster by Institution, Keywords: Analyze which institutions are more likely to research a particular topic.
- Cluster by Advisor, Keywords: Analyze which advisors are more likely to be connected with a particular topic.

Given this information, a set of themes or topics in Information Technology were to be proposed as derived from the data from the repository. These results were to be published to the repository for analyses and comment.

4 RESULTS AND DATA ANALYSIS

As described in Chapter 3, the following methodology was used to collect the data presented in this chapter:

1. Determine a representative sample of theses for a valid study.
 - As many graduate schools as possible were contacted to help with the repository population.
 - These graduate schools were also all asked to provide information on how many master's theses and projects as well as doctoral dissertations they had published in Information Technology.
2. Create and test a repository for graduate level theses and dissertations.
 - A website was created as the mechanism for gathering data.
3. Gather data and populate the repository.
 - All of the graduate publications that had been gathered were entered into the repository.
 - The repository can also be used to edit as well as add any information necessary.
4. Analyze the results, test the hypothesis.
 - A keyword frequency analysis was done on all the research available, and using this data, themes were proposed.

4.1 Gathering data

4.1.1 Contacting schools

Schools with known programs in Information Technology were contacted in order to provide information to populate the repository. Contact information was found from the ABET list of IT accredited schools, recommendations from professors, and from schools contacted recommending other institutions. Each school was asked to provide the following data:

1. Any other graduate programs in Information Technology that should be included in this research.
2. Volunteers to assist with the data collection effort at that school.
3. Information to access the graduate research done at that school.
4. Counts of theses and dissertations that are published each year at that school.

After collecting the above information, and through recommendations given, the following schools were contacted:

Table 4-1: Schools contacted

New Jersey Institute of Technology
BYU-Idaho
BYU-Provo
Georgia Southern University
Capella University
University of Cincinnati
Macon State University
St. John Fisher College
George Mason University
Southern Polytechnic State University
University of Houston
California State University - Channel Islands
Rochester Institute of Technology
Purdue University

Table 4-1: continued

Rensselaer Polytechnic Institute
East Carolina University
Cal State Fullerton
University of Southern Alabama
East Tennessee University
University of Missouri-Kansas City
United States Naval Academy

4.1.2 Determine a representative sample

After contacting the above universities, the following universities responded with information on their active graduate programs:

Table 4-2: Universities with responses

University	Graduates	Online Database	Categorized
Brigham Young University	16 since 2002	http://etd.byu.edu	no
Rochester Institute of Technology	unknown	http://ritdml.rit.edu	very few
Purdue University	unknown	none	n/a
Southern Polytechnic State University	4	none	n/a
Rensselaer Polytechnic Institute	7 since 2002	none	n/a
East Carolina University	unknown	some on ProQuest	no

George Mason University has just begun a new master's program this year and thus does not yet have any graduates.

The process of contacting schools about their graduate programs provided some unexpected results. First of all, finding graduate programs in Information Technology has

proved more difficult than expected. Many of the universities contacted simply did not respond to repeated requests for information, either by e-mail or by phone. Several other questions also arose from those that were contacted.

What constitutes an IT program at a graduate school? There are many schools with Computer Science degrees that appear to contain IT elements to them, but in order to take out bias, only programs clearly (or very close to) Information Technology have been used thus far in the database. However, for example, RIT offers a degree in Information Technology as well as a degree in Networking, Security, and Systems Administration. Both are clearly IT related degrees and the second department is an offshoot of the original IT department, but they are not both specifically labeled as such. A smaller school would probably not have separate departments. Another example is the MS Information Technology that is offered by Capella. Their General Information Technology specialization is clearly IT by title, yet the description of the degree is reminiscent of Information Systems. This is yet another example of confusion in the definition of IT in academia, but as they call themselves an IT program, they must be included as IT research in this study so as not to introduce any researcher bias.

Should master's degrees completed by project instead of by thesis also be included in the repository? Initially, this research just looked at IT *thesis* information, but many graduate programs offer a *project* option either in addition to, or in place of a thesis. The proportion of IT scholarship manifested through projects appears to be significant enough that it would be irresponsible to leave them out. Therefore, the repository needs to have an IT scholarship orientation rather than focus purely on IT theses; however, the ability to distinguish types of documents must be enhanced. This conclusion actually validates the assertion that IT has a significant component that fits Boyer's "scholarship of application" definition (Boyer 1990).

Ekstrom observed that many of the theses in his original two schools reported on development projects (Ekstrom et al. 2006). George Mason's brand new MS Information Technology program is exclusively using projects instead of theses. Rochester Institute of Technology gives the option of either a thesis or a project, however it appears as though the vast majority of their students choose to do projects. Purdue is another university at which a significant number of students choose a project option.

One of the biggest surprises has been common in every school we have worked with thus far. Each department has problems locating their own theses. Most schools just don't normally store their information so as to make past thesis information easily accessible. A school can generally tell how many students graduated in a particular year, and possibly even how many students graduated under an Engineering & Technology umbrella, but not how many students specifically graduated in Information Technology. Each professor has a list of students to whom he has been an advisor, but there has not been an official, publicly available list of every IT thesis in any department that has been involved. It appears that a repository is needed to help institutions track their own IT scholarship.

Even in this age of electronic publishing, not all IT theses are easily available in electronic form. It is surprising how many schools do not yet have Electronic Thesis and Dissertation (ETD) systems. Even when ETD systems are available, there is still the problem of categorization. The theses are often not being indexed by department. This is aggravated by the problem of department naming discussed above. The predicament here is that while the digital thesis is available, how does one know which theses to add to the repository if they are not labeled as IT theses? An individual may pick theses that *look* like IT theses and add them to the repository, but that introduces personal bias into the selection of items defining IT scholarly

work. So far only theses to have come from departments strictly labeled as Information Technology (or very similar such as in the case of RIT) have been added to the repository.

Since it is expected that any thesis currently being written will enjoy the benefit of electronic publishing, one of the primary benefits of the repository may be providing a place for students and faculty to make their work available to others.

4.1.3 Data from individual institutions

The data specific to each school regarding their graduates are as follows:

- **Brigham Young University:** The current graduate committee chair for the IT program has kept a personal list of all the current and previous graduate students. There is no other such list available. According to this list, there have been a total of 16 graduates from the IT program, the first of which graduated in 2002. All of these theses have been added to the repository. BYU has a simple search for all of the electronic theses and dissertations (ETD) from the school, and these theses are all available online.
- **Rochester Institute of Technology:** The number of graduates from the Information Technology program is unknown. There is an online ETD search engine, with 5 theses listed under the Department of Information Technology and another 4 theses listed under Department of Networking, Security, and Systems Administration (both departments under the Computing and Information Sciences school). It is known that RIT has a large number of graduate students, but they were unable to locate all of their publications. The only publications that they could locate and provide were those listed on their ETD page. According to RIT, a large portion of their graduates choose to do projects, which are not published. The 9 theses listed on their ETD page represent all the theses that they have

been able to find in the last several years (2006 is the first one listed). All 9 of these theses are available online and have been added to the repository.

- Purdue: The number of graduates from the Information Technology program is unknown. There is also no online ETD database available, and the only copies they have of theses are by hard copy. Purdue added over 50 publications to the database, the majority of which have been projects. They have done this by getting a list of students that each professor has been the advisor to, looking up the reports in their library, and manually entering each one.
- Southern Polytechnic State University: There have been a total of 4 graduates since 2005. The list of students and thesis titles were provided by a SPSU professor. Basic information on each thesis is available on their library website, but not the full digital copy.
- Rensselaer Polytechnic Institute: There have been a total of 7 graduates since 2002. All 7 theses were found and entered in by a professor at RPI. There is no online database or digital copy of these theses.
- East Carolina University: The number of graduates is unknown. Some of the theses that have been published are accessible through ProQuest. However these theses are not at all categorized, and no list of IT theses could be provided. Therefore none of these have been added to the repository because it is unknown which ones represent IT degrees.

The conclusion is that 200 theses is not a realistic representative sample of graduate publications from IT programs. Of the schools with known numbers of graduates, there are 16 from BYU, 9 from RIT, 4 from SPSU, and 7 from RPI. Therefore a minimum representative sample of publications from five graduate schools is between 35 and 50 rather than around 200.

The main reason this is lower than originally expected is because there are fewer students who graduate with theses than had been expected before this research began. It appears that projects are much more common, and these projects are seldom electronically published. The repository will accept projects as well for the purpose of this research, but they are more difficult to locate and access.

Table 4-3: Thesis distribution by institution

Thesis Distribution by Institution	
Institution	Amount
Purdue University	65
Brigham Young University	16
Rochester Institute of Technology	9
Rensselaer Polytechnic Institute	7
Southern Polytechnic State University	4
Indiana State University	1

Table 4-3 shows the current distribution of graduate publications in the repository. The minimum representative sample has been met, due in good part to the publications from Purdue University. The publication from Indiana State University represents the only PhD dissertation in the database; the rest of the publications are either master's theses or master's projects.

4.2 Google app engine

The advantages of using Google App Engine listed in Chapter 3 have proven to be true. There are, however, some interesting quirks of the service that must be overcome. Many of these quirks show just how new Google App Engine is, and hopefully they will be addressed later on. First of all, there is no method provided by Google to locally backup the database rather

than solely to rely on Google's system. Both the importance of our data as well as the open nature of the repository led to the conclusion that local backups are necessary. There is a third-party backup solution available called Gaebat (Balkan), however it only works with the Django framework whereas we used the Google recommended webapp framework. In order to overcome this, a simple API application was developed to perform local backups. The application used a script. The backup script can query the API script located on Google App Engine for all the information on a particular table. The backup script then parses out all the information and enters it into a locally stored MySQL database.

The API to access the Google App Engine database also shows clear signs of only being in its early stages. Only basic commands are available, but others such as doing a case-insensitive search, joins, counting the items returned, and other commands simply aren't available. This fact also makes the backup API that was created particularly useful: in transferring everything to a MySQL database, all of the SQL commands are available. Finally, a fact that is particularly surprising given Google's search engine origins, there doesn't appear to be any simple way to search the database. This was overcome by the creation of a custom solution using the MySQL database backup.

4.3 User testing of the repository

Limited user testing was done with three groups of people. First academics and students who are familiar with the material in the field were tested as the primary users of this repository. The data entry was also tested with several secretaries as we wanted to be sure that we could use idle times of our administrative staff to accelerate the process and so that the user interface would be generally suitable for administrative staff in the future. Finally, a generic non-technical

group was tested in order to improve the general look and feel of the design. In each case, the subject was instructed to use “talk out-loud protocol”, explaining what he was thinking and any confusion he may have had during the testing. Each subject had a list of thesis titles, and was asked to find the thesis and enter the necessary information into the repository.

The results of the user testing showed that all three groups had similar problem areas. One interesting observation is that not a single person tested actually read the instructions at the top of the “Add a Thesis” page. Each subject just started clicking, which is why the help was moved to a position next to where someone doing data entry would encounter a question, rather than at the top. Changes were made to address all the usability problems that were identified, and all three groups of users experienced greater success after the changes were implemented.

One final observation from the user testing was that the first added document seemed to take a long time, but all of the subsequent documents were finished very quickly. This shows that while there is still a learning curve, it is quickly overcome and documents are easily added thereafter.

4.4 Keyword analysis

Separate word frequency analyses were done on both the thesis documents as well as the citations listed for each document. For the theses, word frequencies were run on the titles, keywords listed, abstracts, and all three put together. Similarly, the citations had a keyword frequencies run on the title, abstracts, and both put together (the citations did not have any keywords listed). Words were listed in groups of 1, 2, and 3 word phrases. In order to facilitate the process, certain commonly used words were removed from the single word list as well as the 2 and 3 word phrases that start with these words. The words taken out were: a, an, and, are, as,

at, be, by, can, for, from, has, have, in, is, of, on, or, our, that, the, these, this, to, was, we, and with. The unfiltered word frequencies are also available.

In the following analyses on each set of word frequencies, each word or word phrase was individually analyzed in the context of its thesis. Therefore conclusions have been made not only solely based on word frequency, but on the word in context. For example, the word ‘it’ could have a high frequency, but depending on the context, the word could be used as the pronoun ‘it’, or the acronym ‘IT’ for Information Technology.

Table 4-4: Title keywords from titles

Thesis Keywords from Titles					
1 word phrases		2 word phrases		3 word phrases	
management	15	project management	6	local area network	2
data	11	analysis of	4	wireless local area	2
information	10	creation of	3	area network implementations	2
security	10	evaluation of	3	network implementations using	2
analysis	10	identity management	3	implementations using real	2
using	10	methodology for	3	selection matrix for	2
project	9	software development	3	solution selection matrix	2
development	9	area network	3	development of a	2
design	9	approach to	3	project management office	2
web	8	analysis and	3	creation of a	2
performance	6	information security	3	security solution selection	2
network	6	framework for	3	using real options	2
software	6	solution selection	2	matrix for wireless	2
system	6	selection matrix	2	performance model of	2
digital	5	choosing a	2	analysis and design	2
computer	5	management for	2	framework for identity	1
framework	5	web services	2	integrated framework for	1
methodology	5	higher education	2	technical support facility	1
wireless	4	design methodology	2	identity management investments	1
building	4	development of	2	support facility ctsf	1
database	4	matrix for	2	practices for managing	1
learning	4	management office	2	bridging the gap	1
identity	4	security and	2	improving project management	1

Table 4-4: continued

1 word phrases		2 word phrases		3 word phrases	
evaluation	4	security solution	2	project management offices	1
assessment	4	real options	2	managing project management	1
technology	4	wireless local	2	tree analysis for	1
approach	4	web application	2	best practices for	1
enterprise	4	design and	2	management at the	1

4.4.1 Thesis keywords from titles

The title keyword list was the best starting point for determining themes because titles are short summaries of the paper topic. Table 4-4 lists the word frequencies used in this analysis. The following phrases emerged as the most relevant.

- **Management (15):** This was a surprising keyword at first glance because Information Technology is not expected to be strongly focused on management (as would be expected for Information Systems), yet it is the highest frequency keyword. The next column helped, however, referring to ‘project management’ (6) and ‘identity management’ (3). A search for the term ‘management’ in the titles further clarifies that the kind of management referred to here is generally not organization management, but management of information, data, resources, etc. Management in IT is clearly interested in using computers to access information more easily.
- **Data (11):** This keyword is to be expected. These are research papers, and they all gather, test, and analyze data. A search showed that the titles refer to data collected. Therefore this keyword is more descriptive of the type of paper (research) rather than a recurring topic in IT.

- Information (10): This keyword can also be largely ignored for the same reasons as data. However, this keyword is important when being used as a description, such as the keyword ‘information security’ (3).
- Security (10): Security is clearly an important part of IT (Lunt et al. 2008). These theses deal with the security of identity, software development, information, and ethics. Worth special mention is ‘identity management’, specifically mentioned in the title by three of the theses in the repository.
- Project (9): Projects are clearly important to IT, but without a description of the project, this keyword is not very useful. ‘Project management’ (6) is the top of the 2 word phrases, which is slightly more descriptive. A search shows theses dedicated mostly towards project management techniques, with all the theses listed as being from Purdue. The Purdue IT research program apparently includes an emphasis on project management techniques.
- Development (9): A search reveals that this keyword focuses primarily on ‘software development’ (3) and the development of systems. This is an important part of IT.
- Web (8): This keyword invariably refers to the Internet and websites. This is also an important part of IT.
- Performance (6): This frequency suggests that IT is worried about performance, but the context of this keyword is scattered. These theses deal with topics ranging from performance of hardware, software, and others. Therefore, IT must be worried about performance of systems in general.
- Network (6): Networking is an important aspect of IT (Lunt et al. 2008). One thing of note is to compare the theses using the ‘web’ (8) keyword with the theses using the

‘network’ (6) keyword. The Internet is essentially a network of computers, but these two keywords generally deal with two different (although related) sets of issues. They should be considered separated themes.

- Software (6): This keyword is often linked with ‘development’ (9). ‘Software development’ (3) is another important part of IT.
- System (6): This keyword used in context represents a collaboration of devices or technology. The frequency of this keyword in all the lists defines IT quite well: Information Technology takes many hardware and software technologies and makes them work together as a system. This keyword does not, however, accurately describe any particular theme within the discipline.

There were also several other keywords of interest:

- Framework (5): This keyword cannot be considered a core teaching of Information Technology, but it certainly appears to be important. Several theses studied frameworks as a means of improving performance and streamlining processes. ‘Framework’ in these theses referred to framework as a general structure.
- Methodology (5): Similar to ‘framework’, this is an important keyword to IT, but relating more to improving performance and streamlining processes.
- Design (9): Also similar to ‘framework’, this keyword refers to using the best possible design both for performance as well as human computer interfacing.
- Database (4): The frequency of this keyword was too low in the analysis of keywords in titles to be considered important from titles alone, but in the analysis of other parts of the documents it shows up more often. Databases are definitely important in Information Technology.

- Learning (4): This keyword appeared surprisingly high, not only in the title frequency list, but in the others as well. A search of this keyword shows that this ‘learning’ does not refer to the constant learning that an IT professional must go through in order to stay current in the field, but rather to learning computer principles and methods of teaching these principles to others. Teaching computer concepts to others appears to be an important part of Information Technology as well.

Table 4-5: Thesis keywords from keywords

Thesis Keywords from Keywords					
1 word phrases		2 word phrases		3 word phrases	
computer	10	human computer	4	human computer interaction	3
web	8	computer interaction	3	computer interaction usability	2
data	7	web application	3	labview data acquisition	1
technology	7	computer security	2	hci labview data	1
information	5	reservation system	2	data acquisition sensors	1
human	4	web services	2	acquisition sensors kitchen	1
cluster	4	interaction usability	2	aware hci labview	1
usability	4	deployment resource	2	interaction context aware	1
management	4	information technology	2	fingerprint social issues	1
application	3	data acquisition	1	biometrics fingerprint social	1
security	3	acquisition sensors	1	social issues technology	1
interaction	3	sensors kitchen	1	computer interaction context	1
computing	3	labview data	1	sensors kitchen table	1
systems	3	aware hci	1	context aware hci	1
design	3	issues technology	1	kitchen table home	1
parallel	3	interaction context	1	artificial intelligence pattern	1
identity	3	context aware	1	ai artificial intelligence	1
fingerprint	2	kitchen table	1	intelligence pattern matching	1
context	2	hci labview	1	pattern matching home	1
interface	2	home context	1	matching home automation	1
website	2	intelligence pattern	1	algorithm ai artificial	1
services	2	artificial intelligence	1	learning algorithm ai	1
reservation	2	pattern matching	1	home context situated	1
eye	2	matching home	1	table home context	1
based	2	automation blank	1	context situated computing	1

Table 4-5: continued

1 word phrases		2 word phrases		3 word phrases	
system	2	home automation	1	situated computing learning	1
database	2	ai artificial	1	computing learning algorithm	1
home	2	algorithm ai	1	environment family technology	1
archiving	2	context situated	1	development environment family	1
quality	2	social issues	1	resource management versioning	1
sorting	2	situated computing	1	deployment resource management	1
overload	2	computing learning	1	management versioning code	1
deployment	2	learning algorithm	1	versioning code versioning	1
biometrics	2	table home	1	versioning cvs svn	1
email	2	biometrics fingerprint	1	code versioning cvs	1
resource	2	cvs svn	1	resource deployment resource	1
development	2	versioning cvs	1	deployment resource deployment	1
versioning	2	svn subversion	1	application multitiered web	1
networking	2	computer music	1	web application multitiered	1
methods	2	paqm nmr	1	multitiered web application	1
programming	2	music paqm	1	web application deployment	1
family	2	code versioning	1	application deployment resource	1
performance	2	versioning code	1	cvs svn subversion	1

4.4.2 Thesis keywords from keywords

Each thesis has listed a title, an abstract, and many have a list of keywords. The “thesis keywords from keywords” is a frequency analysis of all the words typed into the keyword section of each thesis. Not all theses have keywords listed, so the list is small, but still useful. Table 4-5 lists the word frequencies used in this analysis.

This keyword section, as expected, has similar keywords to the title section such as web, web application, security, computer security, and web services. One theme that became apparent from this word frequency list is ‘usability’ (4), ‘computer interaction’ (3), and ‘human computer

interaction' (3). It appears as though usability of a computer plays a role in many of the theses and projects listed.

Table 4-6: Thesis keywords from abstracts

Thesis Keywords from Abstracts					
1 word phrases		2 word phrases		3 word phrases	
data	110	based on	23	one of the	13
system	90	identity management	22	purpose of this	12
information	84	used to	21	some of the	6
study	81	well as	17	based on the	6
it	72	will be	16	digital music device	6
project	65	one of	15	need for a	6
will	60	project management	15	development environment was	5
management	60	purpose of	15	research is to	5
software	58	information technology	15	digital identity management	5
technology	51	web services	14	need to be	5
research	50	order to	14	well as the	5
more	49	use of	13	proof of concept	5
performance	47	it is	12	return on investment	4
used	45	such as	12	iso 17799 standard	4
web	45	number of	12	do not have	4
their	44	study will	11	growth of the	4
computer	44	purdue university	11	electromagnetic field concepts	4
were	42	focus on	11	information technology it	4
using	40	ability to	10	study was to	4
use	40	performance of	10	proposed study will	4
systems	39	need for	10	performance of the	4
security	39	methodology for	9	it project management	4
application	39	set of	9	project management offices	4
students	35	there are	8	enterprise resource planning	3
development	35	need to	8	quality of the	3
process	34	over the	8	level where they	3
learning	34	however the	8	how well a	3
been	34	analysis of	8	due to the	3
which	33	there is	8	quality and reliability	3
one	31	development of	8	analysis of these	3
not	31	united states	8	fingerprint biometric technology	3
based	31	form of	8	such a system	3

Table 4-6: continued

1 word phrases		2 word phrases		3 word phrases	
database	30	effectiveness of	8	identity management solutions	3
design	30	it projects	8	better understanding of	3
tools	29	due to	8	set of guidelines	3
identity	29	quality of	7	course will be	3
developed	29	levels of	7	conceptual understanding of	3
quality	28	using a	7	during the course	3
digital	27	study was	7	will focus on	3
organizations	27	data quality	7	presented to the	3
its	26	digital music	7	significantly higher learning	3
two	26	information systems	7	there is a	3
business	26	understanding of	7	not have a	3
analysis	26	email overload	7	it is the	3
such	26	must be	7	development of the	3
projects	25	computer forensic	7	because of its	3
well	25	information is	7	there are many	3
services	25	design and	7	study focuses on	3
model	25	used in	7	over the last	3
applications	25	development environment	6	development of a	3
course	25	some of	6	physical modeling synthesis	3
many	24	work instructions	6	effectiveness of the	3
methodology	24	technology and	6	quality of service	3
there	24	research is	6	tests will be	3
between	23	proof of	6	objective of this	3
results	23	tool testing	6	confidence and satisfaction	3
provide	23	variety of	6	focus on the	3
also	23	project is	6	large amounts of	3
through	23	through the	6	projects in china	3
they	22	such a	6	forensic tool testing	3
email	22	effects of	6	project was to	3
level	22	best practices	6	digital waveguide synthesis	3
environment	22	data from	6	goal of this	3
each	22	creation of	6	computer forensic tool	3
most	21	part of	6	china and the	3
how	21	music device	6	computer programming instruction	3

4.4.3 Thesis keywords from abstracts

The word frequencies from the abstracts serve to further develop the list of keywords already used. Table 4-6 lists the word frequencies used in this analysis. Once again, some important keywords found are ‘project’ (65), ‘management’ (60), ‘software’ (58), ‘performance’ (47), ‘web’ (45), ‘security’ (39), ‘development’ (35), ‘database’ (30), ‘identity management’ (22), ‘project management’ (15), and ‘web services’ (14). A new keyword that came up is ‘application’ (39) which closely relates to ‘development’.

There were some new results from the three-word phrases. The terms ‘digital music device’, and ‘iso 1799 standard’ each come from a single thesis, so they can be disregarded as a topic from a single research paper rather than a recurring theme. One three-word phrase worth mentioning is ‘proof of concept’. This phrase was mentioned six times by three separate papers. While it is a low frequency, it suggests that IT uses technology to model and scale down larger issues. This can also be another result of the fact that master’s projects and theses are generally proof of concept implementations rather than production systems.

Table 4-7: Thesis keywords from everything

Thesis Keywords from Everything					
1 word phrases		2 word phrases		3 word phrases	
data	128	identity management	26	one of the	13
information	99	based on	24	purpose of this	12
system	98	project management	21	digital music device	7
study	82	used to	21	digital identity management	6
management	79	information technology	19	based on the	6
project	74	web services	18	need for a	6
it	72	well as	17	some of the	6
software	65	will be	16	need to be	5
technology	62	purpose of	15	quality and reliability	5
web	61	one of	15	development environment was	5
will	60	order to	14	iso 17799 standard	5

Table 4-7: continued

1 word phrases		2 word phrases		3 word phrases	
computer	59	use of	13	performance of the	5
performance	55	purdue university	12	development of a	5
security	52	it is	12	research is to	5
research	51	such as	12	proof of concept	5
using	50	methodology for	12	well as the	5
more	49	analysis of	12	project management offices	5
development	46	number of	12	it project management	4
used	45	performance of	11	do not have	4
systems	45	focus on	11	fingerprint biometric technology	4
their	44	study will	11	electromagnetic field concepts	4
application	44	need for	10	security solution selection	4
were	42	development of	10	information technology it	4
design	42	ability to	10	china and the	4
use	40	design and	9	wireless local area	4
learning	39	united states	9	study was to	4
students	37	email overload	9	using real options	4
analysis	37	set of	9	return on investment	4
based	36	creation of	9	solution selection matrix	4
identity	36	information security	8	growth of the	4
database	36	form of	8	proposed study will	4
been	34	there are	8	forensic tool testing	4
process	34	it projects	8	club reservation system	3
which	33	there is	8	enterprise resource planning	3
quality	32	however the	8	tennis club reservation	3
digital	32	data quality	8	agile software development	3
one	31	digital music	8	conceptual understanding of	3
not	31	need to	8	nexrad level ii	3
developed	30	using a	8	confidence and satisfaction	3
course	29	information systems	8	human computer interaction	3
services	29	development environment	8	goal of this	3
tools	29	effectiveness of	8	set of guidelines	3
methodology	29	over the	8	tests will be	3
applications	28	computer forensic	8	presented to the	3
organizations	27	due to	8	it is the	3
model	27	security and	7	life cycle framework	3

Table 4-7: continued

1 word phrases		2 word phrases		3 word phrases	
projects	27	study was	7	identity management solutions	3
business	27	must be	7	site search engine	3
such	26	information is	7	creation of a	3
two	26	levels of	7	due to the	3
its	26	understanding of	7	computer forensic tool	3
network	25	data from	7	such a system	3
well	25	music device	7	computer programming instruction	3
through	25	framework for	7	case based reasoning	3
email	25	used in	7	between china and	3
many	24	quality of	7	mental rotation activities	3
environment	24	identity theft	7	over the last	3
there	24	web application	7	significantly higher learning	3
level	24	tool testing	7	data unit layer	3
between	24	best practices	7	there is a	3
results	23	haptic feedback	6	digital waveguide synthesis	3
provide	23	technology and	6	project management office	3
also	23	project is	6	diskless beowulf clusters	3
testing	22	through the	6	course will be	3
framework	22	quality and	6	objective of this	3
enterprise	22	performance tuning	6	physical modeling synthesis	3
each	22	beowulf clusters	6	identity management solution	3
they	22	model of	6	web site search	3
user	22	software development	6	better understanding of	3
both	21	approach to	6	weather data from	3
tool	21	analysis and	6	projects in china	3
how	21	proof of	6	data from multiple	3
implemented	21	climbing wall	6	network and system	3
solution	21	such a	6	quality of service	3
most	21	information system	6	climbing wall management	3
proposed	20	fingerprint biometric	6	intact transposition of	3
work	20	effects of	6	level where they	3

4.4.4 Thesis keywords from everything

The final word frequency analysis is the thesis keywords from everything. All of the highest word frequencies have already been mentioned, with this final word frequency list confirming the themes that have already emerged. Table 4-7 lists the word frequencies used in this analysis. Several patterns worth looking at emerged after analyzing this list.

- Learning (39): This word continues to have a high frequency, followed closely by 'students' (37). As all of these publications were written by students, it is natural to be interested in helping other students, but this cannot account for all the research found referring to teaching and students. The data leads to the conclusion that currently an important part of research in Information Technology is teaching. This is possibly due to the fact that while most people have to use computers, not everybody understands them as well as IT professionals. Therefore an important part of IT academic research is to be able to explain what you know to other, less computer knowledgeable individuals.
- Database (36): This keyword is not as high as one might expect, yet is still high among all the keyword frequencies. Databases are clearly important to the IT professional, but apparently more as a means to an end rather than an area of study in itself.
- Business (27): Business is not supposed to be as integral a part of Information Technology as, say, Information Systems. Yet it is mentioned somewhere in the title, keyword, or abstract in 19 different publications, which is nearly 20% of the publications gathered. A search of these publications reveals that the majority of

them reference a system or service to be used in a business setting, leading to the conclusion that IT professionals are using IT principles to improve business.

- E-mail (25): This is a new keyword worth mentioning which is clearly related to the 'Internet'.
- Identity management (26): This is at the very top of the 2 word phrases. As mentioned before, this is definitely a subset of security, but clearly a very important one.
- Development environment (8): This is a new keyword that refers to software development.
- Computer forensic (8): This is a new keyword that would be under the category of security.
- 'Data quality' (8), 'quality and reliability' (5): These keywords are more examples of the importance of performance, quality, and accuracy of data in IT research.

4.4.5 Thesis keywords – conclusions

Based on the keyword frequency analyses done, the following themes in IT research have been found:

1. Security: this includes security in anything to do with securing computer hardware and software, with special mention of identity management, ethics, and computer forensics
2. System integration: the identification and/or development of software and hardware to connect components that create a system. This category has several subcategories:
 - Databases
 - Human computer interfacing

3. Networking: physically connecting together computer systems and allowing them to communicate with one another.
4. Internet: Including websites, e-mail, etc.
5. Technical transfer: teaching and learning computing concepts for groups or individuals.

The keyword data showed a fair amount of software development, but not enough database and human computer interfacing to merit separate categories. Networking and Internet on the other hand are both widely studied; yet they are seen as related but separate entities, and thus require separate categories. Finally, the evidence of learning IT principles as a research category doesn't fit any of the other categories, so it merits a category of itself.

Table 4-8: Citations keywords from everything

Citations Keywords from Everything					
1 word phrases		2 word phrases		3 word phrases	
information	215	based on	63	access labs macgyver	23
data	198	use of	57	open access labs	23
system	190	it is	55	labs macgyver principle	23
paper	165	paper we	51	managing open access	23
it	159	such as	47	more with less	17
systems	156	data center	36	doing more with	17
web	143	variety of	35	colleges and universities	16
using	131	computer science	33	universities following the	16
which	129	enterprise management	32	education colleges and	16
user	117	identity theft	30	higher education colleges	16
annotations	115	used to	30	programming philosophies has	16
design	114	using a	29	following the industry	16
computer	108	number of	28	industry trend to	16
use	106	design and	28	trend to decentralize	16
how	105	aspects of	28	departments of higher	16
research	103	order to	25	aspects of enterprise	16
annotation	101	within the	25	philosophies has solidified	16
digital	100	sound synthesis	24	decentralize aspects of	16
such	98	digital waveguide	24	practice within it	16
management	98	set of	23	recent years doing	16

Table 4-8: continued

1 word phrases		2 word phrases		3 word phrases	
used	94	labs macgyver	23	years doing more	16
model	93	access labs	23	reasons in recent	16
applications	93	open access	23	variety of reasons	16
more	92	approach to	23	management to smaller	16
email	90	macgyver principle	23	less has become	16
based	90	managing open	23	become standard operating	16
not	89	information overload	22	within it departments	16
performance	88	used in	22	objectoriented programming philosophies	16
approach	87	analysis of	22	operating practice within	16
users	87	objectoriented programming	21	standard operating practice	16
other	86	using the	21	it departments of	16
synthesis	82	recent years	21	solidified this push	16
results	81	synthesis of	20	enterprise management functions	16
identity	78	well as	19	centralized enterprise management	16
context	76	information retrieval	19	taken from objectoriented	16
computing	75	paper describes	18	functions to individual	16
also	74	identity management	18	departments the use	16
their	73	functions to	18	individual departments the	16
security	71	user interface	18	less centralized enterprise	16
new	71	how to	18	moving less centralized	16
been	69	implementation of	18	functional areas within	16
application	68	taken from	17	smaller functional areas	16
study	67	web pages	17	areas within the	16
contextaware	66	information systems	17	within the organization	16
they	65	more with	17	organization are moving	16
software	62	doing more	17	use of microsofts	16
programming	60	types of	17	management functions to	16
document	60	active directory	17	ou and the	16
analysis	59	management to	17	other object concepts	16
wireless	59	results of	17	enterprise management to	16
both	58	higher education	17	concepts taken from	16
present	58	one of	17	units ou and	16
database	58	centralized enterprise	16	object concepts taken	16

4.4.6 Citations of theses

The current state of the IT thesis repository has less graduate publications than previously hoped for, but there are over 800 research papers cited that are also listed in the database. These research papers are not necessarily all IT research, and this data admittedly is easily skewed by a single thesis having significantly more citations entered than another, but a frequency analysis of these papers is useful to provide more data on what IT researchers are looking at for help and inspiration. Table 4-8 lists the word frequencies used in this analysis.

For the most part, the keywords gathered from the listed citations were very similar to those gathered by the theses themselves (which is to be expected). Keywords that ranked high include 'web', 'design, management', 'applications', 'email', 'performance', and 'data center'. There are some strange keywords that popped up high in the keyword frequencies, but upon further analysis it is all due to a single paper called: *Managing open access labs: "MacGyver principle."* Outside of this, all of the keywords from the citations fit into the suggested themes already listed.

Part of the purpose of collecting the citations is to find common documents referenced by many IT researchers. This was not found, however, as nearly every document in the database was referenced only once, with only a few having been referenced twice. This is in agreement to research that has already been done in the area (Lister and Box 2008).

4.5 Other observations

Part of the data categorized in the repository is what school each publication is from as well as who the main advisor is. It was hypothesized (Ekstrom et al. 2006) that this collection of research may reveal trends specific to what a particular university or a particular professor has

most interest in studying. The database is currently too small to draw strong conclusions, however some initial indications show promise of future understanding of the field, as well as providing a testing ground for the various tools being developed.

The research thus far does not suggest that certain schools produce more papers graduate publications on a particular topic than another. The publications from all schools come from the entire spectrum of IT research. However, only three of the six schools currently represented have over 10 publications in the database. Perhaps a different conclusion can be drawn with additional research.

There is also not enough evidence to conclusively show what each professor's interests are without any prior knowledge about them, but there do seem to be patterns of research with certain advisors. For example, Erick Slazinski of Purdue University has advised several projects on databases and Phillip Rawles, also of Purdue University has advised several papers on wireless technologies. It is very likely that with some prior knowledge of each professor's personal areas of research, patterns of the students they advise will be much more obvious.

4.6 Direct intuitive analysis

The keyword frequency tests appear to be good indicators of themes being researched in IT academia, but in order to test the results, a second direct intuitive analysis was done. For this analysis, the title, keywords, and abstract of each graduate publication was printed out and manually sorted into one of the themes chosen from the keyword frequency lists by the researcher.

The results of the direct intuitive analysis were successful as well as enlightening. Not only did every research paper fit into one of the categories, but nearly every paper could have

easily fit into more than one category. This is also considered a success because of the integrative nature of the IT research domain. For example, software development is not normally the sole topic studied in a paper, but as a means to an end such as the way to connect two systems (which could also be labeled under networking). For the purposes of simple manual categorization, however, only the single best theme was chosen for each publication even when several could have been chosen.

The direct intuitive analysis also brought to light some important subtleties for each of the themes that were chosen. The keyword analysis showed higher frequencies of the terms 'framework', 'quality', 'reliability', and 'methodology'. These were all linked in many of the research papers, some dealing very specifically with using certain frameworks and methodologies to improve the quality and reliability of an end product (such as a software program or the speed of a computer network). This appeared to most closely relate to the system integration category.

Another subtlety is research relating to project management. Several research papers dealt with project management, so clearly this is an important topic in IT. It was decided that the idea of project management belongs under the system integration category. The reason is that project management as a research topic focuses on organizing a team and making informed decisions based on what is understood about the needs of the project. This type of communication falls neatly under system integration.

4.7 Proposed themes in IT

Based on both the keyword frequency analysis as well as the direct intuitive analysis, the following themes for Information Technology are proposed:

1. Security: this includes security in anything to do with securing computer hardware and software, with special mention of identity management, ethics, and computer forensics.
2. System integration: the identification and/or development of software and hardware to connect components that create a system. Worth special mention are frameworks and methodologies for improving quality and reliability. This category has the following subcategories:
 - a. Databases
 - b. Human computer interfacing
 - c. Performance
 - d. Project management
3. Networking: physically connecting together computer systems and allowing them to communicate with one another.
4. Internet: including websites, e-mail, etc.
5. Technical transfer: teaching and learning computing concepts for groups or individuals.

4.8 Comparison to other proposed themes

In 2006, Ekstrom and others (Ekstrom et al. 2006) proposed a tentative set of themes based on 70 master's research papers from two different schools. Their themes are as follows:

1. DEV: Development, building, implementation, integration and other such terms that indicate delivery of a system into a context.
2. ED: These theses came in two flavors, those that were focused on concept learning and those that were actually the application of IT to an educational setting. The

second class could have been considered development, however, we classified them with the ED group because the primary application domain was education.

3. IAS: Information Assurance, Security, and Forensics
4. PROJ: Project management and applications of IT to Project Management. Similar to the ED grouping, we classified development projects focused on the delivery of Project management infrastructure to the PROJ group.
5. TECH: This class includes technology evaluation and testing. “Comparison” and “evaluation” seem to be the most common indicator terms.

These proposed themes proposed by Ekstrom et al closely match in concept to those proposed here, yet the themes proposed here are more tuned in using the indicator terms. For example, there is clearly the need for Network and Internet themes, and while these terms are included in those proposed by Ekstrom et al, the data shows that Network and Internet should stand out as a main theme rather than a smaller sub category of a larger theme.

4.9 Comparison to 5 pillars of IT

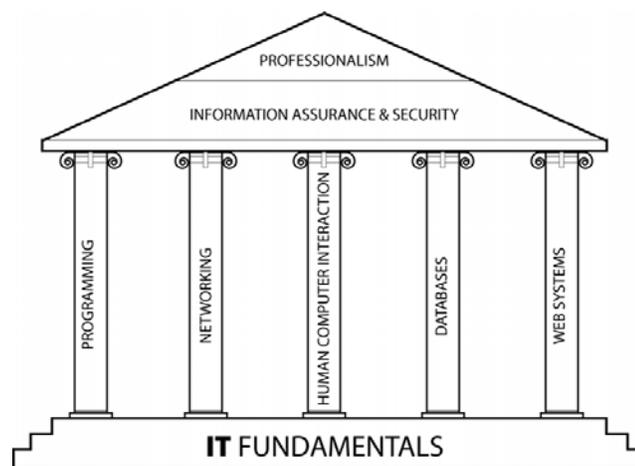


Figure 4-1: Pillars of Information Technology

According to the IT 2008 Curriculum document (Lunt et al. 2008), the pillars of IT are programming, networking, human computer interaction, databases, and web systems. These pillars also have information assurance & security and professionalism overarching them. The proposed themes match very well with these pillars. Networking and web systems match directly with the proposed networking and Internet themes. Programming is the most significant part of the proposed system integration theme, with human computer interaction and databases integrated into that theme. The overarching portion of information assurance and security matches directly with the proposed security theme. Professionalism and IT fundamentals work well with the proposed technical transfer theme.

The proposed themes closely relate to the pillars of Information Technology. This further validates the keyword analysis and proposed themes.

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Overview

Earlier research indicated that there was promise in using theses and dissertations as a window into the themes of IT research, but that a repository was required to successfully pursue the approach. We have implemented a repository and populated it with over 100 documents from 5 graduate programs. An important part of this process was finding out how many theses, projects, and dissertations are published by particular universities in order to create a representative sample for the repository. Finally, a keyword analysis was done on all of the research collected as well as a search function to act as a proof of concept for a cluster analysis.

5.2 Interpretation of findings

The answer to how many publications would constitute a representative sample of IT graduate research was surprisingly low. No school contacted had any easily accessible official record of what theses and dissertations had been published by their IT departments. There were 3 schools, however, that were able to find an exact number, which was surprisingly lower than expected. It is estimated that a representative sample from 5 graduate schools would be around 35 – 50 master's theses and dissertations. Another finding is that many, possibly most, IT students choose a project option rather than a thesis, and therefore projects were added to the repository as well.

The keyword analysis was done both by keyword frequency lists as well as a direct intuitive analysis where each research paper was categorized manually. A set of five themes for IT were proposed using the keywords and tested using the direct analysis. The final themes proposed are as follows:

1. Security: this includes security in anything to do with securing computer hardware and software, with special mention of identity management, ethics, and computer forensics.
2. System integration: the identification and/or development of software and hardware to connect components that create a system. Worth special mention are frameworks and methodologies for improving quality and reliability. This category has the following subcategories:
 - a. Databases
 - b. Human computer interfacing
 - c. Performance
 - d. Project management
3. Networking: physically connecting together computer systems and allowing them to communicate with one another.
4. Internet: including websites, e-mail, etc.
5. Technical transfer: teaching and learning computing concepts for groups or individuals.

5.3 Recommendations for action

The most important thing is to continue to make the repository grow. The more research added, the better the results. All of the analysis tools are online, and the repository can be a helpful tool for anyone interested in Information Technology from an academic perspective.

Another use of the repository is to solve the problem of schools not having any easy means to access their theses and dissertations. If universities upload all of their theses and dissertations to the repository, they will have easy access to know exactly how many people have graduated from their program, what topics they studied, as well as other statistics about advisors, etc.

5.4 Recommendations for further study

Though not exploited in this thesis, the potential exists to use the repository to study the citations, references, and resulting publications of each thesis. Using this information, a web of IT research can be created, perhaps discovering some important base research that many publications cite.

Automation for the repository can also be increased, possibly to include uploading a digital version of a research paper and automatically extracting necessary information.

5.5 Closing

The field of Information Technology is an area of academia that is not going to go away. Research will continue to be published, and as the field grows, it will be important to identify core themes and principles within the discipline to set it apart from others. This research verifies that Information Technology does indeed have a set of core principles that are recognizable from research published within the community of graduate students.

REFERENCES

- Aldrich, H. 1999. *Organizations evolving*. Thousand Oaks: Sage Publications.
- Alford, K. L., Carter, C. A., Ragsdale, D. J., Ressler E. K., and Reynolds , C. W. 2004. Specification and managed development of information technology curricula. In *CITC5 '04: Proceedings of the 5th conference on Information technology education*:261. New York, NY: ACM.
- Balkan, A. Google app engine backup and restore.
- Benbasat, I. and Zmud, R. W. 2003. The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. *MIS Quarterly* 27, no. 2: 183.
- Boyer, E. L. 1990. Scholarship reconsidered.
- Cafarella, E. Doctoral research in educational technology: A directory of dissertations, 1977-2006. <http://cortland.edu/edleadership/edcaffarella/dissdir/>.
- Carroll, J. M. and Roeloffs, R. 1969. Computer selection of keywords using word-frequency analysis. *American Documentation* 20, no. 3: 227-233.
- Denning, P. J. 2001. The profession of it: Who are we? *Commun.ACM* 44, no. 2: 15.
- Ekstrom, J. J., Dark, M. J, Lunt, B. M, and Reichgelt, H. 2006. A research agenda for information technology: Does research literature already exist? In *SIGITE '06: Proceedings of the 7th conference on Information technology education*:19. New York, NY: ACM.
- Ekstrom, J. J. and Lunt, B. 2003. Education at the seams: Preparing students to stitch systems together; curriculum and issues for 4-year it programs. In *CITC4 '03: Proceedings of the 4th conference on Information technology curriculum*:196. New York, NY: ACM.
- Finn, P. J. 1977. Word frequency, information theory, and cloze performance: A transfer feature theory of processing in reading. *Reading Research Quarterly* 13, no. 4: 508-537.
- Google. Why app engine - google app engine - google code. <http://code.google.com/appengine/whyappengine.html> (2009).

- Helps, C. R. G. and Ekstrom, J. J. 2008. Evaluation of a computer networking class in information technology. In *SIGITE '08: Proceedings of the 9th ACM SIGITE conference on Information technology education*:259. New York, NY: ACM.
- Lister, R. and Box, I. 2008. A citation analysis of the sigcse 2007 proceedings. In *SIGCSE '08: Proceedings of the 39th SIGCSE technical symposium on Computer science education*:476. New York, NY: ACM.
- Lunt, B., Ekstrom, J. J., Gorka, S., Hislop, G., Kamali R., Lawson, E., LeBlanc, R., Miller, J. R., and Reichgelt, H.. 2008. Curriculum guidelines for undergraduate degree programs in information technology.
<http://plone.acm.org/education/education/curricula/IT2008%20Curriculum.pdf>.
- Lunt, B. M. and Ekstrom, J. J. 2008. The IT model curriculum: A status update. In *SIGITE '08: Proceedings of the 9th ACM SIGITE conference on Information technology education*:1. New York, NY: ACM.
- Miller, J. R., Gorka, S., Lunt, B., Ekstrom, J. J., Lawson, E., Reichgelt, H., and Kamali, R. 2006. Hiring the IT graduate: What's in the box? In *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations*:6. Washington, DC: IEEE Computer Society.
- Orlikowski, W. J. and Baroudi, J. J. 1991. Studying information technology in organizations: Research approaches and assumptions. *Information Systems Research* 2: 1.
- Pare, G., Bourdeau, S., Marsan, J., Nach, H., and Shuraïda, S. 2008. Re-examining the causal structure of information technology impact research. *European Journal of Information Systems*.
- Reichgelt, H. 2004. Towards a research agenda for information technology. In *CITC5 '04: Proceedings of the 5th conference on Information technology education*:248. New York, NY: ACM.
- Reichgelt, H., Phelps, A., Lunt, B., Slazinski, E., Bruce, C., Ashford, T., and Willis, C. 2004. A comparison of baccalaureate programs in information technology with baccalaureate programs in computer science and information systems. *Journal of Information Technology Education* 3: 1.
- Shackelford, R., McGettrick, A., Sloan, R., Topi, H., Davies, G., Kamali, R., Cross, J., Impagliazzo, J., LeBlanc, R., and Lunt, B. 2006. Computing curricula 2005: The overview report. Paper presented at Proceedings of the 37th SIGCSE technical symposium on Computer science education, Houston, Texas, USA.

APPENDIX A. GRADUATE PUBLICATION KEYWORD LISTS

Thesis Keywords from Titles					
1 word phrases		2 word phrases		3 word phrases	
management	15	project management	6	local area network	2
data	11	analysis of	4	wireless local area	2
information	10	creation of	3	area network implementations	2
security	10	evaluation of	3	network implementations using	2
analysis	10	identity management	3	implementations using real	2
using	10	methodology for	3	selection matrix for	2
project	9	software development	3	solution selection matrix	2
development	9	area network	3	development of a	2
design	9	approach to	3	project management office	2
web	8	analysis and	3	creation of a	2
performance	6	information security	3	security solution selection	2
network	6	framework for	3	using real options	2
software	6	solution selection	2	matrix for wireless	2
system	6	selection matrix	2	performance model of	2
digital	5	choosing a	2	analysis and design	2
computer	5	management for	2	framework for identity	1
framework	5	web services	2	integrated framework for	1
methodology	5	higher education	2	technical support facility	1
wireless	4	design methodology	2	identity management investments	1
building	4	development of	2	support facility ctsf	1
database	4	matrix for	2	practices for managing	1
learning	4	management office	2	bridging the gap	1
identity	4	security and	2	improving project management	1
evaluation	4	security solution	2	project management offices	1
assessment	4	real options	2	managing project management	1
technology	4	wireless local	2	tree analysis for	1
approach	4	web application	2	best practices for	1
enterprise	4	design and	2	management at the	1
solution	3	model of	2	methodology for software	1
biometric	3	using real	2	project methodology for	1
based	3	network	2	climbing wall management	1

		implementations			
applications	3	local area	2	wall management software	1
area	3	implementations using	2	data quality analysis	1
creation	3	information technology	2	project management outcomes	1
enhance	3	performance model	2	heavy project methodology	1
course	3	bridging the	1	analysis and management	1
testing	3	wall management	1	quality analysis and	1
systems	3	management software	1	analysis for light	1
knowledge	3	climbing wall	1	light versus heavy	1
test	2	outcomes from	1	versus heavy project	1
methods	2	data quality	1	central technical support	1
quality	2	management outcomes	1	project management soft	1
simulated	2	improving project	1	study investigating activity	1
utilizing	2	managing project	1	investigating activity theory	1
feedback	2	facility ctsf	1	usability study investigating	1
developing	2	support facility	1	user in the	1
administration	2	management investments	1	emphasizing the user	1
language	2	integrated framework	1	activity theory and	1
through	2	joint projects	1	theory and website	1
media	2	best practices	1	test the influence	1
haptic	2	practices for	1	influence of the	1
implementing	2	quality analysis	1	variance to test	1
impact	2	management at	1	analysis of variance	1
choosing	2	central technical	1	oneway analysis of	1
tracking	2	technical support	1	telecare and telemedicine	1
support	2	management offices	1	concerns in telecare	1
application	2	security concerns	1	between china and	1
forensic	2	variance to	1	china and the	1
visual	2	test the	1	projects between china	1
databases	2	oneway analysis	1	joint projects between	1
networking	2	website navigation	1	outcomes from joint	1
website	2	activity theory	1	decision tree analysis	1
protocol	2	theory and	1	management soft skills	1
engineering	2	influence of	1	staffing a project	1
office	2	network time	1	security concerns in	1
level	2	name service	1	needed for staffing	1
projects	2	service security	1	skills needed for	1
higher	2	domain name	1	soft skills needed	1
engine	2	ntp on	1	management outcomes from	1
clustering	2	time protocol	1	maturity of data	1
standard	2	protocol ntp	1	emerging information technology	1

model	2	investigating activity	1	information technology applications	1
skills	2	study investigating	1	evaluating emerging information	1
personal	2	soft skills	1	approach for evaluating	1
contextaware	2	skills needed	1	organizationalcontext approach for	1
oracle	2	management soft	1	project management life	1
networks	2	united states	1	management life cycle	1
clusters	2	between china	1	network and system	1
encryption	2	china and	1	creating a automatic	1
local	2	needed for	1	framework for network	1
matrix	2	staffing a	1	cycle framework for	1
xml	2	user in	1	life cycle framework	1
students	2	usability study	1	create business value	1
interface	2	emphasizing the	1	development to create	1
practices	2	telecare and	1	rose and requisitepro	1
implementations	2	project methodology	1	using value stream	1
tool	2	concerns in	1	rational rose and	1
creating	2	projects between	1	knowledge of rational	1
options	2	cooperative utilities	1	student knowledge of	1
real	2	cycle framework	1	value stream analysis	1
electronic	2	network and	1	stream analysis the	1
implementation	2	life cycle	1	agile software development	1
measuring	2	management life	1	software development to	1
integration	2	emerging information	1	constraints and agile	1
b2b	2	technology applications	1	theory of constraints	1
graduate	2	system administration	1	analysis the theory	1
value	2	creating a	1	automatic accesslist updating	1
education	2	tool for	1	accesslist updating tool	1
selection	2	cisco ios	1	measuring the impact	1
theory	2	updating tool	1	impact and maturity	1

Thesis Keywords from Keywords					
1 word phrases		2 word phrases		3 word phrases	
computer	10	human computer	4	human computer interaction	3
web	8	computer interaction	3	computer interaction usability	2
data	7	web application	3	labview data acquisition	1
technology	7	computer security	2	hci labview data	1
information	5	reservation system	2	data acquisition sensors	1
human	4	web services	2	acquisition sensors kitchen	1

cluster	4	interaction usability	2	aware hci labview	1
usability	4	deployment resource	2	interaction context aware	1
management	4	information technology	2	fingerprint social issues	1
application	3	data acquisition	1	biometrics fingerprint social	1
security	3	acquisition sensors	1	social issues technology	1
interaction	3	sensors kitchen	1	computer interaction context	1
computing	3	labview data	1	sensors kitchen table	1
systems	3	aware hci	1	context aware hci	1
design	3	issues technology	1	kitchen table home	1
parallel	3	interaction context	1	artificial intelligence pattern	1
identity	3	context aware	1	ai artificial intelligence	1
fingerprint	2	kitchen table	1	intelligence pattern matching	1
context	2	hci labview	1	pattern matching home	1
interface	2	home context	1	matching home automation	1
website	2	intelligence pattern	1	algorithm ai artificial	1
services	2	artificial intelligence	1	learning algorithm ai	1
reservation	2	pattern matching	1	home context situated	1
eye	2	matching home	1	table home context	1
based	2	automation blank	1	context situated computing	1
system	2	home automation	1	situated computing learning	1
database	2	ai artificial	1	computing learning algorithm	1
home	2	algorithm ai	1	environment family technology	1
archiving	2	context situated	1	development environment family	1
quality	2	social issues	1	resource management versioning	1
sorting	2	situated computing	1	deployment resource management	1
overload	2	computing learning	1	management versioning code	1
deployment	2	learning algorithm	1	versioning code versioning	1
biometrics	2	table home	1	versioning cvs svn	1
email	2	biometrics fingerprint	1	code versioning cvs	1
resource	2	cvs svn	1	resource deployment resource	1
development	2	versioning cvs	1	deployment resource deployment	1
versioning	2	svn subversion	1	application multitiered web	1
networking	2	computer music	1	web application multitiered	1
methods	2	paqm nmr	1	multitiered web application	1
programming	2	music paqm	1	web application deployment	1
family	2	code versioning	1	application deployment resource	1
performance	2	versioning code	1	cvs svn subversion	1

Thesis Keywords from Abstracts					
1 word phrases		2 word phrases		3 word phrases	
data	110	based on	23	one of the	13
system	90	identity management	22	purpose of this	12
information	84	used to	21	some of the	6
study	81	well as	17	based on the	6
it	72	will be	16	digital music device	6
project	65	one of	15	need for a	6
will	60	project management	15	development environment was	5
management	60	purpose of	15	research is to	5
software	58	information technology	15	digital identity management	5
technology	51	web services	14	need to be	5
research	50	order to	14	well as the	5
more	49	use of	13	proof of concept	5
performance	47	it is	12	return on investment	4
used	45	such as	12	iso 17799 standard	4
web	45	number of	12	do not have	4
their	44	study will	11	growth of the	4
computer	44	purdue university	11	electromagnetic field concepts	4
were	42	focus on	11	information technology it	4
using	40	ability to	10	study was to	4
use	40	performance of	10	proposed study will	4
systems	39	need for	10	performance of the	4
security	39	methodology for	9	it project management	4
application	39	set of	9	project management offices	4
students	35	there are	8	enterprise resource planning	3
development	35	need to	8	quality of the	3
process	34	over the	8	level where they	3
learning	34	however the	8	how well a	3
been	34	analysis of	8	due to the	3
which	33	there is	8	quality and reliability	3
one	31	development of	8	analysis of these	3
not	31	united states	8	fingerprint biometric technology	3
based	31	form of	8	such a system	3
database	30	effectiveness of	8	identity management solutions	3
design	30	it projects	8	better understanding of	3
tools	29	due to	8	set of guidelines	3
identity	29	quality of	7	course will be	3
developed	29	levels of	7	conceptual understanding of	3
quality	28	using a	7	during the course	3
digital	27	study was	7	will focus on	3

organizations	27	data quality	7	presented to the	3
its	26	digital music	7	significantly higher learning	3
two	26	information systems	7	there is a	3
business	26	understanding of	7	not have a	3
analysis	26	email overload	7	it is the	3
such	26	must be	7	development of the	3
projects	25	computer forensic	7	because of its	3
well	25	information is	7	there are many	3
services	25	design and	7	study focuses on	3
model	25	used in	7	over the last	3
applications	25	development environment	6	development of a	3
course	25	some of	6	physical modeling synthesis	3
many	24	work instructions	6	effectiveness of the	3
methodology	24	technology and	6	quality of service	3
there	24	research is	6	tests will be	3
between	23	proof of	6	objective of this	3
results	23	tool testing	6	confidence and satisfaction	3
provide	23	variety of	6	focus on the	3
also	23	project is	6	large amounts of	3
through	23	through the	6	projects in china	3
they	22	such a	6	forensic tool testing	3
email	22	effects of	6	project was to	3
level	22	best practices	6	digital waveguide synthesis	3
environment	22	data from	6	goal of this	3
each	22	creation of	6	computer forensic tool	3
most	21	part of	6	china and the	3
how	21	music device	6	computer programming instruction	3
both	21	systems and	6	study is to	3
implemented	21	environment was	5	information technology cit	2
work	20	it was	5	system was developed	2
user	20	beowulf clusters	5	information retrieval solutions	2
proposed	20	application development	5	business processes are	2
current	19	designed to	5	detect possible instances	2
tool	19	able to	5	information technology professionals	2
need	19	course of	5	possible instances of	2
three	19	field concepts	5	over the internet	2
network	19	electromagnetic field	5	design and implementation	2
solution	18	digital identity	5	data in a	2

focus	18	solution to	5	students at purdue	2
testing	18	provides a	5	instances of pirated	2
however	18	haptic feedback	5	department of computer	2
organization	18	were used	5	methods used to	2
standard	17	identify the	5	performance difference between	2
available	17	needs of	5	supported by this	2
enterprise	17	information system	5	designed to be	2
implementation	17	study also	5	information in the	2
being	17	climbing wall	5	set of skills	2
personal	17	become a	5	assembly operations and	2
purpose	17	information security	5	mental rotation activities	2
over	17	system was	5	network and system	2
clusters	17	fingerprint biometric	5	system administration projects	2
different	17	research and	5	agile software development	2
framework	17	course will	5	developed for this	2
when	16	identity theft	5	details of the	2
processes	16	performance tuning	5	data analysis algorithms	2
key	16	could be	5	individuals mental rotation	2
paper	16	mental rotation	5	mental rotation ability	2
storage	16	implementation of	5	system developed for	2
requirements	16	consisted of	5	problem solving skills	2
problem	16	demonstrate the	5	system can be	2
provided	16	return on	5	treatment consisted of	2
some	16	through a	5	current state of	2
test	16	fail to	5	study examines the	2
major	16	may be	5	instrument will be	2
issues	16	security and	5	labor cost advantage	2
programming	16	develop a	5	work instruction treatment	2
time	16	goal of	5	according to the	2
integration	16	case study	5	life cycle framework	2
field	16	organizations to	5	asked to perform	2
within	15	focuses on	5	state of data	2
participants	15	access to	5	fail to deliver	2
designed	15	create a	5	collected and stored	2
order	15	open source	5	aid in the	2
while	15	using the	5	solutions are assessed	2
help	15	system and	5	assessed based on	2
components	15	business processes	5	based on level	2
techniques	15	technology it	4	security solutions are	2
into	15	process of	4	candidate security solutions	2
but	15	terms of	4	area networking is	2

ability	14	within a	4	networking is researched	2
new	14	deep freeze	4	level of vulnerability	2
best	14	compared to	4	vulnerability and difficulty	2
set	14	state of	4	real options and	2
number	14	components of	4	options and a	2
university	14	significantly higher	4	security solution selection	2
users	14	process and	4	using real options	2
may	14	management offices	4	evaluated using real	2
specific	14	use in	4	difficulty of use	2
methods	14	physical modeling	4	metrics are evaluated	2
purdue	14	growth of	4	local area networking	2
concept	13	data storage	4	wireless local area	2
professionals	13	quality and	4	purpose of the	2
provides	13	data in	4	increase in the	2
significant	13	it security	4	open source software	2
given	13	lack of	4	open source components	2
solutions	13	developed and	4	study of a	2
deployment	13	deployment process	4	order to provide	2
problems	13	issues in	4	prevent data loss	2
needs	13	context of	4	noise and true	2
survey	13	available for	4	between noise and	2
product	13	family members	4	technology purdue university	2
what	13	areas of	4	web services can	2
effectiveness	13	will allow	4	security of wireless	2
practices	13	software tools	4	many organizations have	2
several	13	developed that	4	application on each	2
value	13	requirements engineering	4	discerning between noise	2
future	12	data integration	4	difficulty discerning between	2
than	12	will help	4	solution selection matrix	2
internet	12	applications and	4	selection matrix is	2
levels	12	provide a	4	however the researcher	2
other	12	user interface	4	provided by the	2
guidelines	12	between the	4	there may be	2
develop	12	it will	4	literature about the	2
created	12	standard for	4	weather data from	2
all	12	learning treatments	4	best practices of	2
support	12	proposed study	4	data from multiple	2
additional	12	more complex	4	nexrad level ii	2
pda	12	web applications	4	introductory programming course	2
tasks	12	life cycle	4	internal database agent	2

technologies	12	designed and	4	implemented as an	2
tests	12	problem solving	4	third of participants	2
become	12	both the	4	wide variety of	2
important	11	software and	4	grow at a	2
compared	11	addition to	4	focuses on the	2
manage	11	resulted in	4	public and private	2
only	11	17799 standard	4	spoken languages and	2
high	11	results from	4	between spoken languages	2
concepts	11	iso 17799	4	implementation guidelines based	2
thesis	11	framework for	4	guidelines based on	2
various	11	which is	4	based on network	2
access	11	system performance	4	provides implementation guidelines	2
area	11	enterprise information	4	software was designed	2
engineering	11	difference between	4	matrix is developed	2
then	11	do not	4	developed that provides	2
first	11	practices and	4	network environment and	2
years	11	not have	4	environment and the	2
cost	11	because of	4	using the hardware accelerated	2
investment	11	since the	4	wireless network on	2
do	11	computer systems	4	technique for each	2
haptic	11	organizations have	4	security technique for	2
traditional	11	solutions to	4	effectiveness of each	2
resources	11	attempt to	4	each security technique	2
department	11	model of	4	one third of	2
no	11	operating system	4	tool testing methodologies	2
overall	10	into the	4	serve as a	2
any	10	study is	4	conclusions of the	2
researcher	10	computer technology	4	components of the	2
including	10	developed to	4	printed circuit board	2
industry	10	means to	4	product with high	2
if	10	applications are	4	it is possible	2
methodologies	10	research has	4	problems that are	2
determine	10	mapping software	4	study are drawn	2
webbased	10	model for	4	web based applications	2
better	10	determine the	4	test and deployment	2
algorithm	10	software was	4	process can be	2
possible	10	sot graduate	4	project management methodologies	2
means	10	amounts of	4	they fail to	2
required	10	development and	4	organizations that have	2

included	10	known as	4	very important to	2
evidence	10	during the	4	technology it projects	2

Thesis Keywords from Everything					
1 word phrases		2 word phrases		3 word phrases	
data	128	identity management	26	one of the	13
information	99	based on	24	purpose of this	12
system	98	project management	21	digital music device	7
study	82	used to	21	digital identity management	6
management	79	information technology	19	based on the	6
project	74	web services	18	need for a	6
it	72	well as	17	some of the	6
software	65	will be	16	need to be	5
technology	62	purpose of	15	quality and reliability	5
web	61	one of	15	development environment was	5
will	60	order to	14	iso 17799 standard	5
computer	59	use of	13	performance of the	5
performance	55	purdue university	12	development of a	5
security	52	it is	12	research is to	5
research	51	such as	12	proof of concept	5
using	50	methodology for	12	well as the	5
more	49	analysis of	12	project management offices	5
development	46	number of	12	it project management	4
used	45	performance of	11	do not have	4
systems	45	focus on	11	fingerprint biometric technology	4
their	44	study will	11	electromagnetic field concepts	4
application	44	need for	10	security solution selection	4
were	42	development of	10	information technology it	4
design	42	ability to	10	china and the	4
use	40	design and	9	wireless local area	4
learning	39	united states	9	study was to	4
students	37	email overload	9	using real options	4
analysis	37	set of	9	return on investment	4
based	36	creation of	9	solution selection matrix	4
identity	36	information security	8	growth of the	4
database	36	form of	8	proposed study will	4
been	34	there are	8	forensic tool testing	4

process	34	it projects	8	club reservation system	3
which	33	there is	8	enterprise resource planning	3
quality	32	however the	8	tennis club reservation	3
digital	32	data quality	8	agile software development	3
one	31	digital music	8	conceptual understanding of	3
not	31	need to	8	nexrad level ii	3
developed	30	using a	8	confidence and satisfaction	3
course	29	information systems	8	human computer interaction	3
services	29	development environment	8	goal of this	3
tools	29	effectiveness of	8	set of guidelines	3
methodology	29	over the	8	tests will be	3
applications	28	computer forensic	8	presented to the	3
organizations	27	due to	8	it is the	3
model	27	security and	7	life cycle framework	3
projects	27	study was	7	identity management solutions	3
business	27	must be	7	site search engine	3
such	26	information is	7	creation of a	3
two	26	levels of	7	due to the	3
its	26	understanding of	7	computer forensic tool	3
network	25	data from	7	such a system	3
well	25	music device	7	computer programming instruction	3
through	25	framework for	7	case based reasoning	3
email	25	used in	7	between china and	3
many	24	quality of	7	mental rotation activities	3
environment	24	identity theft	7	over the last	3
there	24	web application	7	significantly higher learning	3
level	24	tool testing	7	data unit layer	3
between	24	best practices	7	there is a	3
results	23	haptic feedback	6	digital waveguide synthesis	3
provide	23	technology and	6	project management office	3
also	23	project is	6	diskless beowulf clusters	3
testing	22	through the	6	course will be	3
framework	22	quality and	6	objective of this	3
enterprise	22	performance tuning	6	physical modeling synthesis	3
each	22	beowulf clusters	6	identity management solution	3
they	22	model of	6	web site search	3
user	22	software development	6	better understanding of	3
both	21	approach to	6	weather data from	3
tool	21	analysis and	6	projects in china	3
how	21	proof of	6	data from multiple	3

implemented	21	climbing wall	6	network and system	3
solution	21	such a	6	quality of service	3
most	21	information system	6	climbing wall management	3
proposed	20	fingerprint biometric	6	intact transposition of	3
work	20	effects of	6	level where they	3
organization	19	digital identity	6	study is to	3
programming	19	research is	6	networking security and	3
need	19	implementation of	6	not have a	3
three	19	variety of	6	there are many	3
clusters	19	systems and	6	usercentric tabular multicolumn	3
personal	19	open source	6	tabular multicolumn sorting	3
implementation	19	some of	6	because of its	3
standard	19	mental rotation	6	security and systems	3
current	19	develop a	6	how well a	3
test	18	part of	6	project was to	3
focus	18	return on	6	effectiveness of the	3
integration	18	work instructions	6	information retrieval solutions	3
however	18	deep freeze	5	focus on the	3
methods	18	system and	5	quality of the	3
issues	17	enterprise information	5	analysis of these	3
storage	17	data integration	5	performance model of	3
requirements	17	case study	5	study focuses on	3
different	17	information retrieval	5	large amounts of	3
time	17	practices and	5	identity theft victimization	3
available	17	eye tracking	5	will focus on	3
purpose	17	fail to	5	development of the	3
being	17	could be	5	during the course	3
over	17	through a	5	analysis and design	3
within	16	consisted of	5	developed using open	2
field	16	web applications	5	decisions regarding identity	2
some	16	system was	5	open source technologies	2
when	16	demonstrate the	5	increase in the	2
processes	16	field concepts	5	using open source	2
users	16	solution to	5	asked to perform	2
key	16	provides a	5	treatment consisted of	2
provided	16	electromagnetic field	5	enterprise information architecture	2
problem	16	user interface	5	digital repository for	2
paper	16	may be	5	department of computer	2
major	16	requirements engineering	5	work instruction treatment	2
university	15	goal of	5	best practices for	2

order	15	17799 standard	5	practices and procedures	2
best	15	iso 17799	5	business processes are	2
practices	15	focuses on	5	students at purdue	2
value	15	it was	5	according to the	2
but	15	research and	5	resources to provide	2
techniques	15	needs of	5	information technology cit	2
help	15	management offices	5	levels of comprehension	2
while	15	life cycle	5	information systems and	2
components	15	study also	5	advantage of the	2
into	15	data center	5	gaining a better	2
purdue	15	able to	5	varying levels of	2
deployment	15	access to	5	application on each	2
participants	15	development and	5	methodology for building	2
designed	15	identify the	5	demonstrate the application	2
ability	14	were used	5	it is possible	2
wireless	14	web site	5	many organizations have	2
may	14	become a	5	wireless networking equipment	2
new	14	business processes	5	system was developed	2
needs	14	mapping software	5	possible instances of	2
solutions	14	course will	5	telecare and telemedicine	2
specific	14	application development	5	learning modules were	2
survey	14	using the	5	study of a	2
concept	14	environment was	5	students and advising	2
number	14	create a	5	detect possible instances	2
set	14	organizations to	5	it possible to	2
area	14	course of	5	data backup system	2
support	14	designed to	5	distributed data backup	2
problems	13	search engine	4	such as the	2
pda	13	computer technology	4	algorithm across compilers	2
several	13	organizations have	4	java performance of	2
significant	13	known as	4	successful in the	2
provides	13	sensitive data	4	across compilers and	2
professionals	13	resources to	4	instances of pirated	2
interface	13	system for	4	ieee 80211 wireless	2
haptic	13	networking security	4	over the internet	2
usability	13	data in	4	compilers and virtual	2
engineering	13	study is	4	then used to	2
technologies	13	based reasoning	4	data storage needs	2
effectiveness	13	design patterns	4	companies do not	2
product	13	model for	4	use of a	2
what	13	software was	4	design and implementation	2

develop	13	difference between	4	methods used to	2
given	13	human computer	4	sot graduate students	2
investment	12	solutions to	4	implementation of an	2
guidelines	12	security issues	4	web crawler to	2
first	12	will help	4	using web services	2
internet	12	sot graduate	4	other commercial fingerprints	2
created	12	level ii	4	information in the	2
tasks	12	issues in	4	soabased tennis club	2
additional	12	management and	4	purpose of the	2
become	12	amounts of	4	through analysis of	2
family	12	since the	4	finding the right	2
traditional	12	software and	4	been developed and	2
future	12	growth of	4	data in the	2
all	12	data storage	4	software was designed	2
than	12	liberty alliance	4	school of technology	2
department	12	performance model	4	crawler to detect	2
cost	12	during the	4	part of the	2
tests	12	proposed study	4	personal digital assistant	2
other	12	multicolumn sorting	4	digital assistant for	2
algorithm	12	diskless beowulf	4	result of this	2
levels	12	not have	4	management tools and	2
reliability	11	process and	4	were used in	2
assessment	11	choosing a	4	show how well	2
concepts	11	developed and	4	course at purdue	2
important	11	physical modeling	4	focuses on the	2
case	11	projects in	4	wireless network on	2
forensic	11	solution for	4	used to show	2
high	11	it project	4	assembly operations and	2
compared	11	deployment process	4	acceptance of fingerprint	2
thesis	11	software tools	4	current state of	2
knowledge	11	computer systems	4	study examines the	2
various	11	resulted in	4	it projects in	2
skills	11	components of	4	organizations that have	2
music	11	china and	4	system can be	2
evaluation	11	been developed	4	prevent data loss	2
do	11	family members	4	network environment and	2
manage	11	addition to	4	well a student	2
only	11	which is	4	based on network	2
networks	11	many organizations	4	guidelines based on	2
years	11	applications and	4	implementation guidelines based	2
webbased	11	designed and	4	environment and the	2

resources	11	printed circuit	4	effectiveness of each	2
then	11	state of	4	study are drawn	2
no	11	security of	4	collected and stored	2
access	11	both the	4	technique for each	2
sorting	10	results from	4	security technique for	2
biometric	10	biometric technology	4	each security technique	2
developing	10	technology it	4	state of data	2
including	10	use in	4	aid in the	2
overload	10	do not	4	difficulty discerning between	2
overall	10	because of	4	discerning between noise	2
networking	10	within the	4	information technology and	2
any	10	forensic tool	4	technology and relationships	2
industry	10	between the	4	project management methodologies	2
creation	10	problem solving	4	monitor system performance	2
student	10	operating system	4	between noise and	2
required	10	within a	4	details of the	2
create	10	areas of	4	digital asset management	2
material	10	context of	4	data analysis algorithms	2
objective	10	determine the	4	tools used to	2
approaches	10	attempt to	4	noise and true	2
b2b	10	provide a	4	attempt to identify	2
multiple	10	available for	4	rapidly since the	2
effects	10	it security	4	mental rotation ability	2
if	10	system to	4	very important to	2
better	10	system performance	4	offer recommendations for	2
evidence	10	asset management	4	recommendations for a	2
possible	10	more complex	4	systems administration department	2
methodologies	10	evaluation of	4	used to collect	2
file	10	will allow	4	individuals mental rotation	2
resource	10	standard for	4	order to provide	2
approach	10	social engineering	4	computer in order	2
feedback	10	into the	4	project is to	2
included	10	it will	4	there has been	2
graduate	10	computer programming	4	provides implementation guidelines	2
contextaware	10	visual basic	4	developed that provides	2
determine	10	developed to	4	building an eia	2
fingerprint	10	selection matrix	4	tools and techniques	2
means	10	terms of	4	technology purdue university	2
understanding	10	solution selection	4	steps to one	2

education	10	significantly higher	4	included in the	2
researcher	10	waveguide synthesis	4	web services can	2

APPENDIX B. CITATIONS KEYWORD LISTS

Citations Keywords from Everything					
1 word phrases		2 word phrases		3 word phrases	
information	215	based on	63	access labs macgyver	23
data	198	use of	57	open access labs	23
system	190	it is	55	labs macgyver principle	23
paper	165	paper we	51	managing open access	23
it	159	such as	47	more with less	17
systems	156	data center	36	doing more with	17
web	143	variety of	35	colleges and universities	16
using	131	computer science	33	universities following the	16
which	129	enterprise management	32	education colleges and	16
user	117	identity theft	30	higher education colleges	16
annotations	115	used to	30	programming philosophies has	16
design	114	using a	29	following the industry	16
computer	108	number of	28	industry trend to	16
use	106	design and	28	trend to decentralize	16
how	105	aspects of	28	departments of higher	16
research	103	order to	25	aspects of enterprise	16
annotation	101	within the	25	philosophies has solidified	16
digital	100	sound synthesis	24	decentralize aspects of	16
such	98	digital waveguide	24	practice within it	16
management	98	set of	23	recent years doing	16
used	94	labs macgyver	23	years doing more	16
model	93	access labs	23	reasons in recent	16
applications	93	open access	23	variety of reasons	16
more	92	approach to	23	management to smaller	16
email	90	macgyver principle	23	less has become	16
based	90	managing open	23	become standard operating	16
not	89	information overload	22	within it departments	16
performance	88	used in	22	objectoriented programming philosophies	16
approach	87	analysis of	22	operating practice within	16
users	87	objectoriented programming	21	standard operating practice	16

other	86	using the	21	it departments of	16
synthesis	82	recent years	21	solidified this push	16
results	81	synthesis of	20	enterprise management functions	16
identity	78	well as	19	centralized enterprise management	16
context	76	information retrieval	19	taken from objectoriented	16
computing	75	paper describes	18	functions to individual	16
also	74	identity management	18	departments the use	16
their	73	functions to	18	individual departments the	16
security	71	user interface	18	less centralized enterprise	16
new	71	how to	18	moving less centralized	16
been	69	implementation of	18	functional areas within	16
application	68	taken from	17	smaller functional areas	16
study	67	web pages	17	areas within the	16
contextaware	66	information systems	17	within the organization	16
they	65	more with	17	organization are moving	16
software	62	doing more	17	use of microsofts	16
programming	60	types of	17	management functions to	16
document	60	active directory	17	ou and the	16
analysis	59	management to	17	other object concepts	16
wireless	59	results of	17	enterprise management to	16
both	58	higher education	17	concepts taken from	16
present	58	one of	17	units ou and	16
database	58	centralized enterprise	16	object concepts taken	16
two	58	moving less	16	microsofts active directory	16
within	58	less centralized	16	organizational units ou	16
techniques	57	other object	16	directory structure with	16
will	56	object concepts	16	structure with organizational	16
methods	56	organization are	16	active directory structure	16
network	56	concepts taken	16	one of the	14
provide	54	programming philosophies	16	based on the	14
documents	53	solidified this	16	federated identity management	11
than	53	philosophies has	16	paper we describe	10
sound	53	ou and	16	web information systems	10
time	53	units ou	16	sound synthesis of	10
first	52	departments the	16	based on a	9
its	52	individual departments	16	modelbased sound synthesis	9
but	52	microsofts active	16	paper we present	9
between	51	directory structure	16	computer science education	9
technology	51	organizational units	16	paper describes the	8

some	51	structure with	16	some of the	8
framework	50	management functions	16	wireless data sheet	7
different	50	case study	16	it is shown	7
into	50	contextaware applications	16	part of the	7
support	49	practice within	16	wide range of	7
one	48	schema versioning	16	implementation of a	7
interface	48	areas within	16	pacific wireless data	7
enterprise	48	departments of	16	analysis of the	7
modeling	46	it departments	16	research area that	6
what	46	operating practice	16	drive a research	6
java	46	standard operating	16	user interface design	6
most	45	reasons in	16	world wide web	6
environment	45	framework for	16	area that continues	6
quality	44	years doing	16	continues today we	6
personal	44	less has	16	sample of 600	6
many	43	become standard	16	600 mailboxes collected	6
standard	43	education colleges	16	focus on the	6
implementation	43	within it	16	examine a sample	6
work	43	trend to	16	today we examine	6
issues	42	decentralize aspects	16	would drive a	6
less	42	smaller functional	16	term that would	6
while	42	functional areas	16	sidner 8 published	6
services	42	industry trend	16	8 published research	6
important	42	part of	16	ago whittaker and	6
overload	41	colleges and	16	years ago whittaker	6
evaluation	41	following the	16	ten years ago	6
access	41	universities following	16	published research on	6
students	41	there is	15	research on email	6
when	41	performance of	15	synthesis of the	6
models	40	problem of	14	mailboxes collected at	6
through	40	physical modeling	14	coining a term	6
were	40	web services	14	overload coining a	6
open	39	email overload	14	email overload coining	6
theft	39	show that	14	annotations can be	6
internet	39	may be	14	company to compare	6
structure	39	research on	14	archives have grown	6
schema	39	development of	14	there is a	6
number	38	quality of	13	email archives have	6
retrieval	38	how the	13	populations email archives	6
problems	38	systems are	13	1996 our populations	6
center	38	some of	13	grown tenfold we	6

become	37	designed to	13	tenfold we see	6
physical	37	system is	13	distinct strategies for	6
science	37	federated identity	13	strategies for handling	6
recent	37	range of	13	evidence of distinct	6
architecture	37	web information	13	little evidence of	6
well	37	present a	13	see little evidence	6
individual	36	need for	12	size as in	6
variety	36	data centers	12	same size as	6
components	36	personal annotations	12	users organize their	6
development	36	describes the	12	organize their email	6
search	36	approach is	12	how users organize	6
years	35	evaluation of	12	compare how users	6
teaching	34	they are	12	hightech company to	6
there	34	design of	12	their email now	6
method	34	study of	12	email now to	6
departments	34	related to	12	inboxes are roughly	6
education	34	paper presents	11	roughly the same	6
managing	34	according to	11	while inboxes are	6
problem	33	network security	11	1996 while inboxes	6
waveguide	33	system and	11	now to 1996	6
crime	33	data and	11	collected at a	6
about	33	must be	11	whittaker and sidner	6
solutions	33	focus on	11	results show that	6
operating	33	contextaware systems	11	average american internet	6
aspects	33	contextaware computing	11	american internet user	6
concepts	33	overview of	11	internet user is	6
each	33	due to	11	memo the average	6
case	33	concept of	11	data memo the	6
versioning	32	into a	11	data center cooling	6
describes	32	which are	11	center cooling system	6
mobile	32	techniques for	11	purpose of this	6
designed	32	provide a	11	user is not	6
key	32	present the	11	not sure what	6
language	31	introduction to	10	does or what	6
general	31	which is	10	what the term	6
show	31	basis for	10	use of the	6
future	31	context of	10	feed does or	6
fingerprint	31	ability to	10	rss feed does	6
interaction	31	model of	10	sure what podcasting	6
describe	31	survey of	10	what podcasting is	6
string	31	need to	10	podcasting is what	6

order	31	modeling and	10	free data center	6
learning	31	modelbased sound	10	what an rss	6
objectoriented	31	discuss the	10	fargos free data	6
object	30	there are	10	study wells fargos	6
may	30	amount of	10	case study wells	6
parallel	30	show how	10	wells fargos free	6
service	30	annotations are	10	show that the	5
victimization	30	does not	10	digital waveguide mesh	5
set	30	plucked string	10	contextaware applications we	5
distributed	30	annotations on	9	fractional delay filters	5
available	29	where the	9	it is not	5
provides	29	system for	9	model based on	5
devices	29	rather than	9	it can be	5
storage	29	provides a	9	related to the	5
then	29	paper is	9	aspects of the	5
current	29	about the	9	advanced encryption standard	5
developed	29	more than	9	optical disc formats	5
social	28	users to	9	plucked string instruments	5
requirements	28	techniques and	9	paper we propose	5
any	28	information is	9	digital ink annotations	5
higher	28	used for	9	design and implementation	5
objects	28	support the	9	modeling and synthesis	5
presents	28	able to	9	paper presents a	5
high	28	web page	9	paper is to	5
need	28	annotation system	9	physical modeling synthesis	5
active	28	system that	9	realtime sound synthesis	5
environments	27	between the	9	term phishing means	5
networks	27	digital documents	9	need to be	5
functional	27	into the	9	basis for the	5
algorithms	27	applications we	9	nas parallel benchmarks	4
all	27	what the	9	multiple versions of	4
large	27	science education	9	institute of standards	4
discuss	27	management and	9	one or more	4
task	27	methods for	9	standards and technology	4
databases	27	information and	9	national institute of	4
three	27	applications are	9	problem of information	4
people	27	which the	9	teaching and learning	4
practice	26	tool for	9	wireless networked sensors	4
principle	26	string instruments	9	results of the	4
features	26	fractional delay	9	there has been	4
human	26	systems in	8	1d ftd waveguides	4

presented	26	should be	8	part of a	4
various	26	access to	8	techniques that can	4
multimedia	26	databases and	8	c and fortran	4
several	26	do not	8	wide variety of	4
among	26	indicate that	8	presented as a	4
algorithm	26	search engine	8	issues related to	4
clusters	25	revisiting whittaker	8	databases and systems	4
infrastructure	25	model is	8	results indicate that	4
online	25	theory of	8	array of inexpensive	4
efficient	25	principles and	8	based on finite	4
providing	25	supported by	8	such as the	4
prototype	25	digital ink	8	wave propagation in	4
however	25	presents an	8	described and the	4
only	25	systems the	8	information present in	4
part	25	characteristics of	8	focuses on the	4
same	25	banner ads	8	used in the	4
where	24	results are	8	usability performance and	4
labs	24	attention to	8	data center design	4
communication	24	class of	8	implementation of the	4
over	24	comparison of	8	point for the	4
text	24	pacific wireless	8	information about the	4
life	24	user is	8	digital waveguide modeling	4
middleware	24	guide to	8	2d gabor wavelets	4
theory	24	high performance	8	redundant array of	4
industry	24	performance and	8	social security number	4
	1	such a	8	nature of the	4
building	24	user interfaces	8	computer science courses	4
report	24	describe a	8	performance and maintainability	4
control	24	terms of	8	starting point for	4
related	24	compared to	8	paper we discuss	4
common	23	research and	8	used as a	4
				software development	
course	23	information about	8	environments	4
tools	23	document management	7	extent to which	4
functions	23	methods are	7	paper presents an	4
privacy	23	wide range	7	how to build	3
existing	23	applications and	7	algorithm of rc4	3
content	23	optical disc	7	serial programming howto	3
strategies	23	semantic web	7	scheduling algorithm of	3
survey	23	digital annotation	7	ways in which	3
macgyver	23	annotation of	7	information retrieval solutions	3
organizational	22	through the	7	paper reviews the	3

field	22	problems of	7	first step in	3
them	22	qualitative research	7	approach to the	3
public	22	description of	7	federated databases and	3
studies	22	deal with	7	study of the	3
biometric	22	first year	7	principles and practice	3
audio	22	information to	7	high performance computing	3
do	22	results show	7	design and evaluation	3
banner	22	how these	7	real time on	3
writing	22	but also	7	present the design	3
understanding	22	how users	7	industrial electronic monitors	3
areas	22	propose a	7	web search engine	3
victims	22	changes to	7	input and output	3
single	22	importance of	7	application design and	3
pages	22	annotation systems	7	enhancements to the	3
types	22	requirements for	7	such as a	3
usability	22	way to	7	key scheduling algorithm	3
shared	21	purpose of	7	according to the	3
so	21	support for	7	retina based identification	3
organization	21	representation of	7	synthesis of plucked	3
federated	21	finally we	7	operator interface terminals	3
traditional	21	email archives	7	international telecommunication union	3
approaches	21	years ago	7	teaching object orientation	3
technical	21	content and	7	teaching objectoriented programming	3
way	21	impact of	7	problem of teaching	3
propose	21	than the	7	concurrent versions system	3
form	21	wireless data	7	compared to the	3
much	21	lack of	7	objectoriented design and	3
terms	21	while the	7	how it can	3
multiple	21	systems that	7	life expectancy of	3
principles	21	optical discs	7	characteristics of the	3
discussion	21	parts of	7	design and programming	3
processing	21	personal information	7	applied to the	3
units	21	associated with	7	implementation of an	3
activities	21	computer crime	7	deal with the	3
characteristics	21	effects of	7	goal is to	3
optical	20	focused on	7	large number of	3
must	20	annotation and	7	message passing interface	3
focus	20	1d fdfd	7	feelings of email	3
made	20	social security	7	crime rate trends	3
described	20	information technology	7	describes the implementation	3

instruments	20	form of	7	identity theft victimization	3
guide	20	synthesis is	7	also be used	3
interfaces	20	environment for	7	it provides a	3
designing	20	applied to	7	audit trail service	3
discussed	20	kinds of	7	such information is	3
community	20	overload and	7	rely heavily on	3
taken	20	will be	7	system based on	3
electronic	20	strategies for	7	present in hypertext	3
should	20	presents a	7	amount of information	3

APPENDIX C. GOOGLE APP ENGINE CODE

App.yaml

```
application: it-thesis
version: 3b
runtime: python
api_version: 1

handlers:

- url: /css
  static_dir: css

- url: /images
  static_dir: images

- url: /js
  static_dir: js

- url: /
  script: home.py

- url: /backup/*
  script: backup.py

- url: /(*\.(html))
  script: home.py

- url: /*
  script: redirect.py
```

backup.py

```
import sys
import cgi
import datetime
```

```

from google.appengine.ext import db
from google.appengine.api import users
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
from google.appengine.ext.db import polymodel

class Institution(db.Model):
    name=db.StringProperty()
    city=db.StringProperty()
    state=db.StringProperty()
    zip=db.StringProperty()
    country=db.StringProperty()
    user_add=db.UserProperty(auto_current_user_add=True)
    user_modify = db.UserProperty(auto_current_user=True)

class Person(db.Model):
    lastname=db.StringProperty()
    firstname=db.StringProperty()
    middlename=db.StringProperty()
    email=db.EmailProperty()
    address=db.PostalAddressProperty()
    user_add=db.UserProperty(auto_current_user_add=True)
    user_modify = db.UserProperty(auto_current_user=True)

class Affiliation(db.Model):
    institution=db.ReferenceProperty(Institution)
    person=db.ReferenceProperty(Person)
    affiliation=db.StringProperty(choices=["Faculty: Full
Professor",
"Faculty: Assistant Professor",
"Faculty: Associate Professor",
"Faculty: Adjunct",
"Student: MS",
"Student: Ph.D.",
"Student: UnderGrad"])
    user_add=db.UserProperty(auto_current_user_add=True)
    user_modify = db.UserProperty(auto_current_user=True)

class Document(polymodel.PolyModel):
    title=db.StringProperty()
    #author= db.ListProperty(db.Key)
    abstract=db.TextProperty()
    url = db.TextProperty()
    pubyear = db.StringProperty()
    keywords = db.StringProperty()

```

```

        modifydate=db.DateTimeProperty(auto_now_add=True)
        user_add=db.UserProperty(auto_current_user_add=True)
        user_modify = db.UserProperty(auto_current_user=True)

class GenericDoc(Document):
    author= db.ListProperty(db.Key)
    bibtex = db.TextProperty()

class Thesis(Document):
    doctype=db.StringProperty(choices=[ "Thesis",

"Dissertation",

"Project"])
    author=
db.ReferenceProperty(Person,collection_name="secondmodel_referen
ce_one_set")
    institution = db.ReferenceProperty(Institution)
    advisor =
db.ReferenceProperty(Person,collection_name="secondmodel_referen
ce_two_set")
    otheradvisors = db.ListProperty(db.Key)
    pubs_citations = db.ListProperty(db.Key)
    pubs_refs = db.ListProperty(db.Key)
    pubs_resulting = db.ListProperty(db.Key)

delim = ' ### '
newdelim = '###\n\n###'

class ThesisListHandler(webapp.RequestHandler):
    def get(self):
        output = ''
        list = Thesis.all()
        for item in list:
            abstract = item.abstract
            if not abstract:
                abstract = ''
            url = item.url
            if not url:
                url = ''
            pubyear = item.pubyear
            if not pubyear:
                pubyear = ''
            keywords = item.keywords
            if not keywords:

```

```

        keywords = ''
doctype = item.doctype
if not doctype:
    doctype = ''
author = str(item.author.key())
if not author:
    author = ''
institution = str(item.institution.key())
if not institution:
    institution = ''
advisor = str(item.advisor.key())
if not advisor:
    advisor = ''
otheradvisors = ''
for a in item.otheradvisors:
    otheradvisors += str(a)+' , '
otheradvisors = ''
for a in item.otheradvisors:
    otheradvisors += str(a)+' , '
pubs_citations = ''
for a in item.pubs_citations:
    pubs_citations += str(a)+' , '
pubs_refs = ''
for a in item.pubs_refs:
    pubs_refs += str(a)+' , '
pubs_resulting = ''
for a in item.pubs_resulting:
    pubs_resulting += str(a)+' , '
user_add = ''
if item.user_add:
    user_add = str(item.user_add.user_id())
user_modify = ''
if item.user_modify:
    user_modify =
str(item.user_modify.user_id())
user_nick = ''
if item.user_add:
    user_nick = str(item.user_add.nickname())
user_nick_mod = ''
if item.user_modify:
    user_nick_mod =
str(item.user_modify.nickname())
modifydate = ''
if item.modifydate:
    modifydate = str(item.modifydate)
output += str(item.key()+delim
output += item.title+delim

```

```

        output += abstract+delim
        output += url+delim
        output += pubyear+delim
        output += keywords+delim
        output += author+delim
        output += doctype+delim
        output += institution+delim
        output += advisor+delim
        output += otheradvisors+delim
        output += pubs_citations+delim
        output += pubs_refs+delim
        output += pubs_resulting+delim
        output +=
user_add+delim+user_modify+delim+user_nick+delim+user_nick_mod+d
elim

        output += modifydate
        output += newdelim
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write(output)

class DocumentListHandler(webapp.RequestHandler):
    def get(self):
        output = ''
        list = GenericDoc.all()
        for item in list:
            abstract = item.abstract
            if not abstract:
                abstract = ''
            url = item.url
            if not url:
                url = ''
            pubyear = item.pubyear
            if not pubyear:
                pubyear = ''
            keywords = item.keywords
            if not keywords:
                keywords = ''
            author = ''
            for a in item.author:
                author += str(a)+', '
            bibtex = item.bibtex
            if not bibtex:
                bibtex = ''
            user_add = ''
            if item.user_add:
                user_add = str(item.user_add.user_id())
            user_modify = ''

```

```

        if item.user_modify:
            user_modify =
str(item.user_modify.user_id())
            user_nick = ''
            if item.user_add:
                user_nick = str(item.user_add.nickname())
            user_nick_mod = ''
            if item.user_modify:
                user_nick_mod =
str(item.user_modify.nickname())
            modifydate = ''
            if item.modifydate:
                modifydate = str(item.modifydate)
            output += str(item.key()+delim
            output += item.title+delim
            output += abstract+delim
            output += url+delim
            output += pubyear+delim
            output += keywords+delim
            output += author+delim
            output += bibtex+delim
            output +=
user_add+delim+user_modify+delim+user_nick+delim+user_nick_mod+d
elim

            output += modifydate
            output += newdelim
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write(output)

class AffiliationListHandler(webapp.RequestHandler):
    def get(self):
        output = ''
        list = Affiliation.all()
        for item in list:
            affil = str(item.institution.key())
            if not affil:
                affil = ''
            person = str(item.person.key())
            if not person:
                person = ''
            user_add = ''
            if item.user_add:
                user_add = str(item.user_add.user_id())
            user_modify = ''
            if item.user_modify:
                user_modify =
str(item.user_modify.user_id())

```

```

        output +=
str(item.key()+delim+affil+delim+person+delim+item.affiliation+
delim
        output += user_add+delim+user_modify
        output += newdelim
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write(output)

```

```

class InstitutionListHandler(webapp.RequestHandler):
    def get(self):
        output = ''
        list = Institution.all()
        for item in list:
            name = item.name
            if not name:
                name = ''
            city = item.city
            if not city:
                city = ''
            state = item.state
            if not state:
                state = ''
            zip = item.zip
            if not zip:
                zip = ''
            country = item.country
            if not country:
                country = ''
            user_add = ''
            if item.user_add:
                user_add = str(item.user_add.user_id())
            user_modify = ''
            if item.user_modify:
                user_modify =
str(item.user_modify.user_id())
            output +=
str(item.key()+delim+name+delim+city+delim+state+delim+zip+deli
m+country+delim
            output += user_add+delim+user_modify
            output += newdelim
            self.response.headers['Content-Type'] = 'text/plain'
            self.response.out.write(output)

```

```

class PersonListHandler(webapp.RequestHandler):
    def get(self):
        output = ''
        list = Person.all()

```

```

for item in list:
    mid = item.middlename
    if not mid:
        mid = ''
    email = item.email
    if not email:
        email = ''
    user_add = ''
    if item.user_add:
        user_add = str(item.user_add.user_id())
    user_modify = ''
    if item.user_modify:
        user_modify =
str(item.user_modify.user_id())
    output +=
str(item.key()+delim+item.lastname+delim+item.firstname+delim+m
id+delim+email+delim
    output += user_add+delim+user_modify
    output += newdelim
self.response.headers['Content-Type'] = 'text/plain'
self.response.out.write(output)

application = webapp.WSGIApplication(
    [
        ('/backup/person',PersonListHandler),

('/backup/institution',InstitutionListHandler),

('/backup/affiliation',AffiliationListHandler),

('/backup/document',DocumentListHandler),

        ('/backup/thesis',ThesisListHandler)],
        debug=True)

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()

```

redirect.py

```
import sys
import cgi
import datetime
import wsgiref.handlers
import os
import urllib;

from google.appengine.ext import db
from google.appengine.api import users
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
from google.appengine.ext.webapp import template
from google.appengine.ext.db import polymodel

defaultRedirect = "base"

a_types = ["Faculty: Full Professor",
           "Faculty: Assistant Professor",
           "Faculty: Associate Professor",
           "Faculty: Adjunct",
           "Student: MS",
           "Student: Ph.D.",
           "Student: UnderGrad"]
doc_types = ["Thesis",
             "Dissertation",
             "Project"]

class Institution(db.Model):
    name=db.StringProperty()
    city=db.StringProperty()
    state=db.StringProperty()
    zip=db.StringProperty()
    country=db.StringProperty()
    user_add=db.UserProperty(auto_current_user_add=True)
    user_modify = db.UserProperty(auto_current_user=True)

class Person(db.Model):
    lastname=db.StringProperty()
    firstname=db.StringProperty()
    middlename=db.StringProperty()
    email=db.EmailProperty()
    address=db.PostalAddressProperty()
    user_add=db.UserProperty(auto_current_user_add=True)
    user_modify = db.UserProperty(auto_current_user=True)
```

```

class Affiliation(db.Model):
    institution=db.ReferenceProperty(Institution)
    person=db.ReferenceProperty(Person)
    affiliation=db.StringProperty(choices=["Faculty: Full
Professor",

"Faculty: Assistant Professor",

"Faculty: Associate Professor",

"Faculty: Adjunct",

"Student: MS",

"Student: Ph.D.",

"Student: UnderGrad"])
    user_add=db.UserProperty(auto_current_user_add=True)
    user_modify = db.UserProperty(auto_current_user=True)

class Document(polymodel.PolyModel):
    title=db.StringProperty()
    #author= db.ListProperty(db.Key)
    abstract=db.TextProperty()
    url = db.TextProperty()
    pubyear = db.StringProperty()
    keywords = db.StringProperty()
    modifydate=db.DateTimeProperty(auto_now_add=True)
    user_add=db.UserProperty(auto_current_user_add=True)
    user_modify = db.UserProperty(auto_current_user=True)

class GenericDoc(Document):
    author= db.ListProperty(db.Key)
    bibtex = db.TextProperty()

class Thesis(Document):
    doctype=db.StringProperty(choices=["Thesis",

"Dissertation",

```

```

"Project" ])
    author=
db.ReferenceProperty(Person, collection_name="secondmodel_referen
ce_one_set")
    institution = db.ReferenceProperty(Institution)
    advisor =
db.ReferenceProperty(Person, collection_name="secondmodel_referen
ce_two_set")
    otheradvisors = db.ListProperty(db.Key)
    pubs_citations = db.ListProperty(db.Key)
    pubs_refs = db.ListProperty(db.Key)
    pubs_resulting = db.ListProperty(db.Key)

# DocumentList is not currently being used
class DocumentList(db.Model):
    docset=db.ReferenceProperty(Document)
    ReferenceType=db.StringProperty(choices=[ "Citation",
                                                "Reference",
                                                "Related Result" ])

#-----
# Institution form handler
#-----

class InstitutionFormHandler(webapp.RequestHandler):
    def get(self):
        output = ""
        <h2>Enter a new Institution</h2>
        <form method="POST" action="/Institution">
        <table border="0">
        <tr><td>Name:</td><td><input type="text" name="name"
size="40"></td></tr>
        <tr><td>City:</td><td><input type="text" name="city"
size="40"></td></tr>
        <tr><td>State:</td><td><input type="text" name="state"
size="40"></td></tr>
        <tr><td>Zip:</td><td><input type="text" name="zip"
size="40"></td></tr>
        <tr><td>Country:</td><td><input type="text"
name="country" size="40"></td></tr>
        <tr><td>&nbsp;</td><td><input type="submit"></td></tr>
        </table>
        </form>

```

```

        """
        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
            'templates/generic.html')
        self.response.out.write(template.render(path,
            template_values))

    def post(self):

        output = '<h2>Posted new Institution</h2>'

        data = Institution()
        data.name = self.request.get('name')
        data.city = self.request.get('city')
        data.state = self.request.get('state')
        data.zip = self.request.get('zip')
        data.country = self.request.get('country')

        data.put()
        output += 'success'

        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
            'templates/generic.html')
        self.response.out.write(template.render(path,
            template_values))

class InstitutionListHandler(webapp.RequestHandler):
    def get(self):
        query = db.GqlQuery("SELECT * FROM Institution ORDER
BY name")
        output = """
        <link rel="stylesheet" href="/css/thesislist.css"
type="text/css" />
        <table border="1" width="100%" class="thesislist">

            <tr><td>Name</td><td>City</td><td>State</td><td>Zip</td><td>
>Country</td></tr>

```

```

"""
for instit in query:
    if (instit.city):
        city = instit.city
    else:
        city = ''
    if (instit.zip):
        zip = instit.zip
    else:
        zip = ''
    if (instit.country):
        country = instit.country
    else:
        country = ''
    output += "<tr>"
    output += '<td><a
href="/InstitutionEdit?id='+str(instit.key())+' ">'+instit.name+'
</a></td>'
    output += "<td>"+city+"</td>"
    output += "<td>"+instit.state+"</td>"
    output += "<td>"+zip+"</td>"
    output += "<td>"+country+"</td>"
    output += "</tr>"
output += """
</table>
"""
template_values = {
    'bodytext': output,
    'logintext': logintext,
}

path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
self.response.out.write(template.render(path,
template_values))

class InstitutionEditHandler(webapp.RequestHandler):
    def get(self):
        if not user:
            output = 'In order to access this page, we ask
that you please <a href="%s">login</a>.' %
users.create_login_url(self.request.uri)
        else:
            instit = db.get(db.Key(self.request.get('id')))
            if (instit.city):
                city = instit.city
            else:

```

```

        city = ''
    if (instit.zip):
        zip = instit.zip
    else:
        zip = ''
    if (instit.country):
        country = instit.country
    else:
        country = ''
    output = ''
    output += """
<h2>Edit an Institution</h2>
<form method="POST" action="/InstitutionEdit">
"""
    output += '<input type="hidden" name="id"
value="' + self.request.get('id') + '" />'
    output += '<table border="0">'
    output += '<tr><td>Name:</td><td><input
type="text" name="name" size="40"
value="' + instit.name + '"></td></tr>'
    output += '<tr><td>City:</td><td><input
type="text" name="city" size="40" value="' + city + '"></td></tr>'
    output += '<tr><td>State:</td><td><input
type="text" name="state" size="40"
value="' + instit.state + '"></td></tr>'
    output += '<tr><td>Zip:</td><td><input
type="text" name="zip" size="40" value="' + zip + '"></td></tr>'
    output += '<tr><td>Country:</td><td><input
type="text" name="country" size="40"
value="' + country + '"></td></tr>'
    output += """
<tr><td>&nbsp;</td><td><input
type="submit"></td></tr>
</table>
</form>
"""
    template_values = {
        'bodytext': output,
        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

def post(self):

```

```

if not user:
    sys.exit()
output = '<h2>Edited Institution</h2>'

error = None
if not self.request.get('name'):
    error = 'You must enter the name.'
if not self.request.get('state'):
    error = 'You must enter the state.'

if error == None:
    data = db.get(db.Key(self.request.get('id')))
    data.name = self.request.get('name').strip()
    if (self.request.get('city') != None):
        data.city = self.request.get('city').strip()
    else:
        data.city = None
    data.state = self.request.get('state').strip()
    if (self.request.get('zip')):
        data.zip = self.request.get('zip').strip()
    if (self.request.get('country')):
        data.country =
self.request.get('country').strip()

    data.put()
    output += 'Thank you for editing. <a
href="/InstitutionList">Back to list</a>'
    if error != None:
        output += 'There was an error: '+error

    template_values = {
        'bodytext': output,
        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

#-----
# Insert form handler
#-----

class InsertHandler(webapp.RequestHandler):

```

```

def get(self):
    #for i in (self.request.get):
    #    i = i.strip()
    error = None
    if self.request.get('type') == 'person':
        data = Person()
        if not self.request.get('firstname').strip():
            error = 'You must enter a first name.'
        if not self.request.get('lastname').strip():
            error = 'You must enter a last name.'
        #if not error: # check to make sure not adding
same thing twice

        if not error:
            exists = Person.all()
            if self.request.get('firstname').strip():
                data.firstname =
self.request.get('firstname').strip()
                exists.filter("firstname = ",
self.request.get('firstname').strip())
                if self.request.get('middlename').strip():
                    data.middlename =
self.request.get('middlename').strip()
                    exists.filter("middlename = ",
self.request.get('middlename').strip())
                    if self.request.get('lastname').strip():
                        data.lastname =
self.request.get('lastname').strip()
                        exists.filter("lastname = ",
self.request.get('lastname').strip())
                        if self.request.get('email').strip():
                            data.email =
self.request.get('email').strip()
                            exists.filter("email = ",
self.request.get('email').strip())

                            exists2 = exists.fetch(1000)

                            key = None
                            if len(exists2) < 1:
                                data.put()
                                key = data.key()

                            else: # loop through list to check they are
exactly the same
                                for q in exists2:

```

```

        if q.middlename == data.middlename
and q.email == data.email:
            key = q.key()
            continue
        if not key:
            data.put()
            key = data.key()

        #Person.get_or_insert(data)
        #key = data.key

        if self.request.get('institution') and
self.request.get('affiliation'):
            newdata = Affiliation()
            exists = Affiliation.all()
            if self.request.get('institution'):
                newdata.institution =
db.Key(self.request.get('institution'))
                exists.filter("institution = ",
db.Key(self.request.get('institution')))
            if self.request.get('affiliation'):
                newdata.affiliation =
self.request.get('affiliation')
                exists.filter("affiliation = ",
self.request.get('affiliation'))
            newdata.person = key
            exists.filter("person = ", key)
            if not exists.get():
                newdata.put()

        json = "{\n"
        json += '\t"result": "success",\n'
        json += '\t"key": "%s",\n' % str(key)
        json += '\t"firstname": "%s",\n' %
data.firstname
        json += '\t"middlename": "%s",\n' %
(data.middlename)
        json += '\t"lastname": "%s",\n' %
data.lastname
        json += '\t"email": "%s"\n' % data.email
        json += '}'

        self.response.out.write(json);
        #self.response.out.write('did it and wrote
'+data.lastname)
    else:

```

```

        json = "{\n"
        json += '\t"result": "fail",\n'
        json += '\t"error": "%s"\n' % error
        json += '}'

        self.response.out.write(json);
elif self.request.get('type') == 'institution':
    data = Institution()
    exists = Institution.all()

    if not self.request.get('name').strip():
        error = 'You must enter the name.'
    if not self.request.get('state').strip():
        error = 'You must enter the state.'

    if not error:
        data.name = self.request.get('name').strip()
        exists.filter("name = ",
self.request.get('name').strip())
        if self.request.get('city').strip():
            data.city =
self.request.get('city').strip()
            data.state =
self.request.get('state').strip()
            exists.filter("state = ",
self.request.get('state').strip())
            if self.request.get('zip').strip():
                data.zip =
self.request.get('zip').strip()
                if self.request.get('country').strip():
                    data.country =
self.request.get('country').strip()

        exists2 = exists.get()

        if not exists2:
            data.put()
            key = data.key()
        else:
            key = exists2.key()

        json = "{\n"
        json += '\t"result": "success",\n'
        json += '\t"key": "%s",\n' % str(key)
        json += '\t"name": "%s",\n' % data.name
        json += '\t"city": "%s",\n' % (data.city)
        json += '\t"state": "%s",\n' % data.state

```

```

        json += '\t"zip": "%s",\n' % data.zip
        json += '\t"country": "%s"\n' % data.country
        json += '}'

        self.response.out.write(json);
else:
    json = "{\n"
    json += '\t"result": "fail",\n'
    json += '\t"error": "%s"\n' % error
    json += '}'

    self.response.out.write(json);

elif self.request.get('type') == 'doc_prev':
    data = db.get(db.Key(self.request.get('doc')))
    firstauthor = Person().get(data.author[0])
    json = "{\n"
    json += '\t"result": "success",\n'
    json += '\t"key": "%s",\n' %
str(self.request.get('doc'))
    json += '\t"title": "%s",\n' % data.title
    json += '\t"author": "%s",\n' %
(firstauthor.firstname+' '+firstauthor.lastname)
    json += '\t"year": "%s"\n' % data.pubyear
    json += '}'

    self.response.out.write(json)
def post(self):
    error = None
    if self.request.get('type') == 'doc':
        if not self.request.get('title').strip():
            error = 'You must enter a title.'
        if not self.request.get('author').strip():
            error = 'You must enter at least 1 author.'

    if not error:
        authors =
self.request.get('author').split('|')

        for i in range(len(authors)):
            authors[i] = db.Key(authors[i])

        data = GenericDoc()
        data.title = self.request.get('title')
        data.author = authors
        if self.request.get('abstract'):

```

```

        data.abstract =
urllib.unquote(self.request.get('abstract'))
        if self.request.get('url').strip():
            data.url =
self.request.get('url').strip()
            if self.request.get('year').strip():
                data.pubyear =
self.request.get('year').strip()
            if self.request.get('bibtex').strip():
                data.bibtex =
urllib.unquote(self.request.get('bibtex').strip())

        data.put()
        key = data.key()

        firstauthor = Person().get(data.author[0])

        json = "{\n"
        json += '\t"result": "success",\n'
        json += '\t"key": "%s",\n' % str(key)
        json += '\t"title": "%s",\n' % data.title
        json += '\t"author": "%s",\n' %
(firstauthor.firstname+' '+firstauthor.lastname)
        json += '\t"year": "%s"\n' % data.pubyear
        json += '}'

        self.response.out.write(json)
    else:
        json = "{\n"
        json += '\t"result": "fail",\n'
        json += '\t"error": "%s"\n' % error
        json += '}'

        self.response.out.write(json);

    if self.request.get('type') == 'thesis':
        if not self.request.get('title'):
            error = 'You must enter a title.'
        if not self.request.get('author'):
            error = 'You must enter the author.'

    if not error:
        advisorlist =
self.request.get('key_advisor').split('|')
        citationlist =
self.request.get('key_citations').split('|')

```

```

        reflist =
self.request.get('key_references').split('|')
        publist =
self.request.get('key_publications').split('|')

        data = Thesis()
        data.title = self.request.get('title')
        data.author =
db.Key(self.request.get('author'))
        data.institution =
db.Key(self.request.get('institution'))
        if self.request.get('advisor'):
            data.advisor =
db.Key(self.request.get('advisor'))
            if self.request.get('key_advisor'):
                for i in range(len(advisorlist)):
                    advisorlist[i] =
db.Key(advisorlist[i])
                    data.otheradvisors = advisorlist
                    if self.request.get('keywords').strip():
                        data.keywords =
self.request.get('keywords').strip()
                        if self.request.get('abstract'):
                            data.abstract =
urllib.unquote(self.request.get('abstract'))
                            if self.request.get('url').strip():
                                data.url =
self.request.get('url').strip()
                                if self.request.get('year'):
                                    data.pubyear = self.request.get('year')
                                if self.request.get('year'):
                                    data.doctype =
self.request.get('doctype')
                                if self.request.get('key_citations'):
                                    for i in range(len(citationlist)):
                                        citationlist[i] =
db.Key(citationlist[i])
                                        data.pubs_citations = citationlist
                                        if self.request.get('key_references'):
                                            for i in range(len(reflist)):
                                                reflist[i] = db.Key(reflist[i])
                                            data.pubs_refs = reflist
                                        if self.request.get('key_publications'):
                                            for i in range(len(publist)):
                                                publist[i] = db.Key(publist[i])
                                            data.pubs_resulting = publist

```

```

        data.put()
        key = data.key()

    output = 'Thank you for adding this thesis.'

    if error:
        output = 'Your thesis was not entered, there
was an error with the submission. Please go back and try again.'

    template_values = {
        'bodytext': output,
        'logintext': logintext,
    }
    path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

class getInfoHandler(webapp.RequestHandler):
    def get(self):
        output = ''
        if self.request.get('type') == 'person':
            query = db.GqlQuery("SELECT * FROM Person ORDER
BY lastname")
            for person in query:
                output += '<option value="%s">%s,
%s</option>' %
(str(person.key()),person.lastname,person.firstname)
                self.response.out.write(output)
            elif self.request.get('type') == 'institution':
                query = db.GqlQuery("SELECT * FROM Institution
ORDER BY name")
                for institution in query:
                    output += '<option value="%s">%s</option>' %
(str(institution.key()),institution.name)
                    self.response.out.write(output)

#-----
# Thesis form handler
#-----

class ThesisFormHandler(webapp.RequestHandler):
    def get(self):
        logmein = ''

```

```

        if not user:
            logmein = '<font color="red">While it is not a
requirement, it is preferrable that you <a href="%s">login</a>
before entering information to give you credit for your
work.</font><br /><br />' % users.create_login_url('/Thesis')

            list_institution = db.GqlQuery("SELECT * FROM
Institution ORDER BY name")
            list_person = db.GqlQuery("SELECT * FROM Person ORDER
BY lastname")
            now = datetime.datetime.now()
            list_years = range(1970,now.year+1)
            list_docs = GenericDoc.gql("ORDER BY title")

            post_to = '/insert'
            template_values = {
                'post_to': post_to,
                'logmein': logmein,
                'a_types': a_types,
                'doc_types': doc_types,
                'list_docs': list_docs,
                'logintext': logintext,
                'list_institution': list_institution,
                'list_person': list_person,
                'list_years': list_years,
            }

            path = os.path.join(os.path.dirname(__file__),
'templates/thesis.html')
            self.response.out.write(template.render(path,
template_values))

class ThesisListHandler(webapp.RequestHandler):
    def get(self):

        list = Thesis.gql("ORDER BY title")

        template_values = {
            'logintext': logintext,
            'list_thesis': list,
        }

        path = os.path.join(os.path.dirname(__file__),
'templates/thesis_list.html')
        self.response.out.write(template.render(path,
template_values))

```

```

class ThesisEditHandler(webapp.RequestHandler):
    def get(self):
        if not user:
            output = 'In order to access this page, we ask
that you please <a href="%s">login</a>.' %
users.create_login_url(self.request.uri)
            template_values = {
                'bodytext': output,
                'logintext': logintext,
            }

            path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
            self.response.out.write(template.render(path,
template_values))
        else:
            doc = db.get(db.Key(self.request.get('id')))
            if (doc.abstract):
                abstract = doc.abstract
            else:
                abstract = ''
            if (doc.keywords):
                keywords = doc.keywords
            else:
                keywords = ''
            if (doc.url):
                url = doc.url
            else:
                url = ''
            post_to = '/ThesisEdit'
            doc_institution = doc.institution.key()
            doc_title = doc.title
            doc_abstract = abstract
            doc_keywords = keywords
            doc_advisor = str(doc.advisor.key())
            doc_url = url
            doc_year = str(doc.pubyear)

            output = ''
            otherhiddenfields = '<input type="hidden"
name="id" value="'+self.request.get('id')+' " />'
            list_institution = db.GqlQuery("SELECT * FROM
Institution ORDER BY name")
            list_person = db.GqlQuery("SELECT * FROM Person
ORDER BY lastname")
            now = datetime.datetime.now()

```

```

list_years = range(1970,now.year+1)
list_docs = GenericDoc.gql("ORDER BY title")

output += '<script language="Javascript">'
for i in doc.otheradvisors:
    advisor = Person().get(i)
    name = advisor.lastname+', '
    if (advisor.middlename):
        name += advisor.firstname+'
'+advisor.middlename
    else:
        name += advisor.firstname
    output +=
'autosuggest_addItem_advisor("'" +str(advisor.key())+"',
"+" +name+""); '
        for i in doc.pubs_citations:
            pub = GenericDoc().get(i)
            firstauthor = Person().get(pub.author[0])
            name = firstauthor.lastname+', '
            if (firstauthor.middlename):
                name += firstauthor.firstname+'
'+firstauthor.middlename
            else:
                name += firstauthor.firstname
            output +=
'addItem("citation", "'" +str(pub.key())+"', "' +pub.title+"',
"+" +name+""); '
                for i in doc.pubs_refs:
                    pub = GenericDoc().get(i)
                    firstauthor = Person().get(pub.author[0])
                    name = firstauthor.lastname+', '
                    if (firstauthor.middlename):
                        name += firstauthor.firstname+'
'+firstauthor.middlename
                    else:
                        name += firstauthor.firstname
                    output +=
'addItem("reference", "'" +str(pub.key())+"', "' +pub.title+"',
"+" +name+""); '
                        for i in doc.pubs_resulting:
                            pub = GenericDoc().get(i)
                            firstauthor = Person().get(pub.author[0])
                            name = firstauthor.lastname+', '
                            if (firstauthor.middlename):
                                name += firstauthor.firstname+'
'+firstauthor.middlename
                            else:

```

```

        name += firstauthor.firstname
        output +=
'addItem("publication",'+str(pub.key())+', "'+pub.title+',
'+name+'); '

        advisorname = doc.advisor.lastname+',
'+doc.advisor.firstname
        if (doc.advisor.middlename):
            advisorname += ' '+doc.advisor.middlename
        authorname = doc.author.lastname+',
'+doc.author.firstname
        if (doc.author.middlename):
            authorname += ' '+doc.author.middlename
        output +=
'pickFromList('+str(doc.advisor.key())+', "'+advisorname+',
"advisor"); '
        output +=
'pickFromList('+str(doc.author.key())+', "'+authorname+',
"author"); '

        output += """
</script>
"""

        list_years2 = []
        b = 0
        for a in list_years:
            list_years2.append(str(a))
            b = b + 1

        template_values = {
            'output': output,
            'post_to': post_to,
            'a_types': a_types,
            'doc_types': doc_types,
            'list_docs': list_docs,
            'logintext': logintext,
            'list_institution': list_institution,
            'list_person': list_person,
            'list_years': list_years2,
            'doc_institution': doc_institution,
            'doc_title': doc_title,
            'doc_abstract': doc_abstract,
            'doc_keywords': doc_keywords,
            'doc_advisor': doc_advisor,
            'doc_url': doc_url,
            'doc_year': doc_year,
            'otherhiddenfields': otherhiddenfields,

```

```

    }

    path = os.path.join(os.path.dirname(__file__),
                        'templates/thesis.html')
    self.response.out.write(template.render(path,
    template_values))

    def post(self):
        if not user:
            sys.exit()
        error = None
        data = db.Key(self.request.get('id'))
        output = '<h2>Edited Thesis</h2>'
        if not self.request.get('title').strip():
            error = 'You must enter a title.'
        if not self.request.get('author'):
            error = 'You must enter the author.'

        if not error:
            advisorlist =
self.request.get('key_advisor').split('|')
            citationlist =
self.request.get('key_citations').split('|')
            reflist =
self.request.get('key_references').split('|')
            publist =
self.request.get('key_publications').split('|')

            data.title = self.request.get('title').strip()
            data.author = db.Key(self.request.get('author'))
            data.institution =
db.Key(self.request.get('institution'))
            if self.request.get('advisor'):
                data.advisor =
db.Key(self.request.get('advisor'))
            else:
                data.advisor = None
            if self.request.get('key_advisor'):
                for i in range(len(advisorlist)):
                    advisorlist[i] = db.Key(advisorlist[i])
                data.otheradvisors = advisorlist
            else:
                data.otheradvisors = []
            if self.request.get('keywords').strip():
                data.keywords =
self.request.get('keywords').strip()
            else:

```

```

        data.keywords = None
    if self.request.get('abstract'):
        data.abstract =
urllib.unquote(self.request.get('abstract'))
    else:
        data.abstract = None
    if self.request.get('url').strip():
        data.url = self.request.get('url').strip()
    else:
        data.url = None
    if self.request.get('year'):
        data.pubyear = self.request.get('year')
    else:
        data.pubyear = None
    if self.request.get('doctype'):
        data.doctype = self.request.get('doctype')
    else:
        data.doctype = None
    if self.request.get('key_citations'):
        for i in range(len(citationlist)):
            citationlist[i] =
db.Key(citationlist[i])
            data.pubs_citations = citationlist
    else:
        data.pubs_citations = []
    if self.request.get('key_references'):
        for i in range(len(reflist)):
            reflist[i] = db.Key(reflist[i])
        data.pubs_refs = reflist
    else:
        data.pubs_refs = []
    if self.request.get('key_publications'):
        for i in range(len(publist)):
            publist[i] = db.Key(publist[i])
        data.pubs_resulting = publist
    else:
        data.pubs_resulting = []

    data.put()
    output += 'Thank you for editing. <a
href="/ThesisList">Back to list</a>'

    if error != None:
        output += 'There was an error: '+error

    template_values = {
        'bodytext': output,

```

```

        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

#-----
-----
# Person form handler
#-----
-----

class PersonFormHandler(webapp.RequestHandler):
    def get(self):
        output = """
        <h2>Enter a new Person</h2>
        <form method="POST" action="/Person">
        <table border="0">
        <tr><td>First Name:</td><td><input type="text "
name="firstname" size="40"></td></tr>
        <tr><td>Middle Name:</td><td><input type="text "
name="middlename" size="40"></td></tr>
        <tr><td>Last Name:</td><td><input type="text "
name="lastname" size="40"></td></tr>
        <tr><td>E-mail Address:</td><td><input type="text "
name="email" size="40"></td></tr>
        <tr><td>Address:</td><td><input type="text "
name="address" size="40"></td></tr>
        <tr><td>&nbsp;</td><td><input type="submit"></td></tr>
        </table>
        </form>
        """
        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
        self.response.out.write(template.render(path,
template_values))

    def post(self):

        output = '<h2>Posted new Person</h2>'

```

```

        data = Person()
        data.firstname = self.request.get('firstname').strip()
        data.middlename =
self.request.get('middlename').strip()
        data.lastname = self.request.get('lastname').strip()
        data.email = self.request.get('email').strip()
        data.address = self.request.get('address').strip()

        data.put()
        output += 'success'

        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
        self.response.out.write(template.render(path,
template_values))

```

```

class PersonListHandler(webapp.RequestHandler):
    def get(self):
        output = ""
        <link rel="stylesheet" href="/css/thesislist.css"
type="text/css" />
        <table border="1" width="100%" class="thesislist">
        <tr><td>Name</td><td>Affiliations</td></tr>
        ""
        list = Person.all()
        for guy in list:
            output += "<tr>"
            output += "<td><a
href=\" /PersonEdit?id="+str(guy.key())+"\">"+guy.firstname+ "
            if (guy.middlename != None):
                output += guy.middlename+ "
            output += guy.lastname+"</td>"
            output += "<td>"
            blah = db.GqlQuery("SELECT * FROM Affiliation
WHERE person = :1 LIMIT 3", guy)
            affilcount = 0
            for affil in blah:
                if (affilcount >= 1):

```

```

        output += ', '
        affilcount = affilcount + 1
        output += affil.institution.name + "(" +
affil.affiliation+)"
        output += "</td>"
        output += "</tr>"
    output += ""
</table>
"""
    template_values = {
        'bodytext': output,
        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

class PersonEditHandler(webapp.RequestHandler):
    def get(self):
        if not user:
            output = 'In order to access this page, we ask
that you please <a href="%s">login</a>.' %
users.create_login_url(self.request.uri)
        else:
            person = db.get(db.Key(self.request.get('id')))
            output = ''
            if (self.request.get('action') == 'delaffil'):

                db.delete(db.Key(self.request.get('affilID')))
            elif (self.request.get('action') == 'addaffil'):
                if self.request.get('institutionID') and
self.request.get('affiliation'):
                    newdata = Affiliation()
                    exists = Affiliation.all()
                    if self.request.get('institutionID'):
                        newdata.institution =
db.Key(self.request.get('institutionID'))
                        exists.filter("institution = ",
db.Key(self.request.get('institutionID')))
                    if self.request.get('affiliation'):
                        newdata.affiliation =
self.request.get('affiliation')
                        exists.filter("affiliation = ",
self.request.get('affiliation'))
                    newdata.person = person

```

```

        exists.filter("person = ", person)
        if not exists.get():
            newdata.put()
# I keep out personal information such as email
or address

    if (person.middlename != None):
        mid = person.middlename
    else:
        mid = ''
    output += """
    <h2>Edit a Person</h2>
    <form method="POST" action="/PersonEdit">
    """
    output += '<input type="hidden" name="id"
value="' + self.request.get('id') + '" />'
    output += '<table border="0">'
    output += '<tr><td>First Name:</td><td><input
type="text" name="firstname" size="40"
value="' + person.firstname + '"></td></tr>'
    output += '<tr><td>Middle Name:</td><td><input
type="text" name="middlename" size="40"
value="' + mid + '"></td></tr>'
    output += '<tr><td>Last Name:</td><td><input
type="text" name="lastname" size="40"
value="' + person.lastname + '"></td></tr>'
    output += """
    <tr><td valign="top">E-mail
Address:</td><td><input type="text" name="email" size="40"><br
/*Not shown for privacy</td></tr>
    <tr><td valign="top">Address:</td><td><input
type="text" name="address" size="40"><br /*Not shown for
privacy</td></tr>
    <tr><td>&nbsp;</td><td><input
type="submit"></td></tr>
    </table>
    </form>
    <br /><br />
    <h2>Affiliations</h2>
    <table border="1" cellpadding="2" cellspacing="0"
width="500">

    <tr><td>Institution</td><td>Affiliation</td><td>&nbsp;</td></tr>
    """
    for a in Affiliation.gql("WHERE person = :1",
person):

```

```

        output +=
'<tr><td>'+a.institution.name+'</td><td>'+a.affiliation+'</td>'
        output += '<td><a
href="/PersonEdit?action=delaffil&affilID='+str(a.key())+'&id='+
self.request.get('id')+' ">Remove</a>'
        output += '</tr>'
    output += "</table>"
    output += """
<h2>Enter a new Affiliation</h2>
<form method="GET" action="/PersonEdit">
<input type="hidden" name="id" value="%s" />
<input type="hidden" name="action"
value="addaffil" />
<table border="0">
<tr><td>Institution:</td><td><select
name="institutionID">
    """ % self.request.get('id')
    query = db.GqlQuery("SELECT * FROM Institution
ORDER BY name")
    for person in query:
        output += '<option value="%s">%s</option>' %
(str(person.key()),person.name)
    output += '</select></td></tr>'
    output += '<tr><td>Affiliation
Type:</td><td><select name="affiliation">'
    for a in a_types:
        output += '<option value="%s">%s</option>' %
(a,a)
    output += """
</select>
</td></tr>
<tr><td>&nbsp;</td><td><input
type="submit"></td></tr>
"""
    output += '</table></form>'

    template_values = {
        'bodytext': output,
        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

def post(self):

```

```

if not user:
    sys.exit()
output = '<h2>Edited Person</h2>'

error = None
if not self.request.get('firstname').strip():
    error = 'You must enter a first name.'
if not self.request.get('lastname').strip():
    error = 'You must enter a last name.'

if error == None:
    data = db.get(db.Key(self.request.get('id')))
    data.firstname = self.request.get('firstname')
    if (self.request.get('middlename').strip()):
        data.middlename =
self.request.get('middlename').strip()
    else:
        data.middlename = None
    data.lastname =
self.request.get('lastname').strip()
    if (self.request.get('email').strip()):
        data.email =
self.request.get('email').strip()
    if (self.request.get('address').strip()):
        data.address =
self.request.get('address').strip()

    data.put()
    output += 'Thank you for editing. <a
href="/PersonList">Back to list</a>'
    if error != None:
        output += 'There was an error: '+error

    template_values = {
        'bodytext': output,
        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

#-----
# Affiliation form handler

```

```

#-----
-----

class AffiliationFormHandler(webapp.RequestHandler):
    def get(self):
        output = ""
        <h2>Enter a new Affiliation</h2>
        <form method="POST" action="/Affiliation">
        <table border="0">
        <tr><td>Institution:</td><td><select
name="institutionID">
        ""
        query = db.GqlQuery("SELECT * FROM Institution ORDER
BY name")
        for person in query:
            output += '<option value="%s">%s</option>' %
(str(person.key()),person.name)
            output += '</select></td></tr>'
            output += '<tr><td>Person:</td><td><select
name="personID">'
            query = db.GqlQuery("SELECT * FROM Person ORDER BY
lastname")
            for person in query:
                output += '<option value="%s">%s %s</option>' %
(str(person.key()),person.firstname,person.lastname)
                output += '</select></td></tr>'

            output += '<tr><td>Affiliation Type:</td><td><select
name="affiliation">'
            for a in a_types:
                output += '<option value="%s">%s</option>' %
(a,a)

            output += ""
            </select>
            </td></tr>
            <tr><td>&nbsp;</td><td><input type="submit"></td></tr>
            ""
            output += '</table></form>'

        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')

```

```

        self.response.out.write(template.render(path,
template_values))

    def post(self):
        output = '<h2>Posted new Affiliation</h2>'

        data = Affiliation()
        data.institutionID =
db.Key(self.request.get('institutionID'))
        data.personID = db.Key(self.request.get('personID'))
        data.affiliation = self.request.get('affiliation')

        data.put()
        output += 'success'

        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
        self.response.out.write(template.render(path,
template_values))

class AffiliationListHandler(webapp.RequestHandler):
    def get(self):
        output = ""
        <link rel="stylesheet" href="/css/thesislist.css"
type="text/css" />
        <table border="1" width="100%" class="thesislist">

        <tr><td>Institution</td><td>Person</td><td>Affiliation</td>
</tr>
        ""
        list = Affiliation.all()
        for affil in list:
            output += "<tr>"
            output += "<td>"+affil.institution.name+"</td>"
            output += "<td>"+affil.person.lastname+"</td>"
            output += "<td>"+affil.affiliation+"</td>"
            output += "</tr>"
        output += ""
        </table>
        ""
        template_values = {

```

```

        'bodytext': output,
        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
                        'templates/generic.html')
    self.response.out.write(template.render(path,
template_values))

#-----
# Document form handler
#-----

class DocumentListHandler(webapp.RequestHandler):
    def get(self):
        query = GenericDoc.gql("ORDER BY title")

        output = """
        <link rel="stylesheet" href="/css/thesislist.css"
type="text/css" />
        <table border="1" width="100%" class="thesislist">
        <tr><td>Title</td><td>Author</td><td>Publication
Year</td><td>&nbsp;</td></tr>
        """
        for doc in query:
            author = Person().get(doc.author[0])
            if not doc.pubyear:
                pubyear = ''
            else:
                pubyear = doc.pubyear
            output += "<tr>"
            output += '<td><a
href="/DocumentEdit?id='+str(doc.key())+' ">'+doc.title+'</a></td
>'

            output += "<td>"+author.lastname+"</td>"
            output += "<td>"+pubyear+"</td>"
            output += "<td>"+doc.class_name()+"</td>"
            output += "</tr>"
        output += """
        </table>
        """
        template_values = {
            'bodytext': output,

```

```

        'logintext': logintext,
    }

    path = os.path.join(os.path.dirname(__file__),
        'templates/generic.html')
    self.response.out.write(template.render(path,
        template_values))

class DocumentEditHandler(webapp.RequestHandler):
    def get(self):
        if not user:
            output = 'In order to access this page, we ask
that you please <a href="%s">login</a>.' %
users.create_login_url(self.request.uri)
            template_values = {
                'bodytext': output,
                'logintext': logintext,
            }

            path = os.path.join(os.path.dirname(__file__),
        'templates/generic.html')
            self.response.out.write(template.render(path,
        template_values))
        else:
            output = ''
            doc = db.get(db.Key(self.request.get('id')))

            if (doc.abstract):
                abstract = doc.abstract
            else:
                abstract = ''
            if (doc.url):
                url = doc.url
            else:
                url = ''

            output += """
                <script type="text/javascript"
src="/js/docauthorlist.js"></script>
                <link href="/js/doclist.css" rel="stylesheet"
type="text/css" />
                <h2>Edit a Document</h2>
                <form method="POST" action="/DocumentEdit"
name="adddoc">
                """
            output += '<input type="hidden" name="id"
value="' + self.request.get('id') + '" />'

```

```

        output += '<table border="0">'
        output += '<tr><td>Title:</td><td><input
type="text" name="title" size="40"
value="' + doc.title + '"></td></tr>'
        output += ""
        <tr>
            <td>Authors:</td>
            <td>
                <div id='docauthorlist'><select
name="author">
                    <option value="">Select Person</option>
                    ""
                    for person in db.GqlQuery("SELECT * FROM Person
ORDER BY lastname"):
                        output += '<option value="%s">%s,
%s</option>' %
(str(person.key()),person.lastname,person.firstname)
                    output += ""
                </select>
                <input type="button" value="Add Author
to Document" onClick="insertAuthor();"></div>
                <div id='docauthormultilist'></div>
            </td>
        </tr>
        ""
        output += '<tr><td>Abstract:</td><td><textarea
name="abstract" rows="4"
cols="40">' + abstract + '</textarea></td></tr>'
        output += '<tr><td>URL:</td><td><input
type="text" name="url" size="40" value="' + url + '"></td></tr>'
        output += '<tr><td>Year
Published:</td><td><select name="year"><option value="">Select a
Year</option>'
        now = datetime.datetime.now()
        yearlist = range(1970,now.year+1)
        for i in yearlist:
            if str(i) == doc.pubyear:
                temp = ' selected'
            else:
                temp = ''
            output += "<option value='%s'%s>%s</option>"
% (i,temp,i)
        output += ""
        </select></td></tr>
        <tr><td>&nbsp;</td><td><input type="submit"
value="Submit"></td></tr>
        </table>

```

```

        </form>
        <script
language="Javascript">onLoadFunction_docauthor();</script>
        <script language="Javascript">
        """
        for i in doc.author:
            author = Person().get(i)
            name = author.lastname+', '
            if (author.middlename):
                name += author.firstname+'
'+author.middlename
            else:
                name += author.firstname
            output +=
'addItem_docauthor("'+str(author.key())+'", "'+name+'"); '
            output += """
        </script>
        """

        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
        self.response.out.write(template.render(path,
template_values))

    def post(self):
        if not user:
            sys.exit()
        output = '<h2>Edited Document</h2>'

        error = None
        if not self.request.get('title').strip():
            error = 'You must enter a title.'
        if not self.request.get('key_docauthor'):
            error = 'You must enter at least 1 author.'

        if error == None:
            authors =
self.request.get('key_docauthor').split('|')
            data = db.get(db.Key(self.request.get('id')))
            data.title = self.request.get('title').strip()
            data.pubyear = self.request.get('year')
            for i in range(len(authors)):

```

```

        authors[i] = db.Key(authors[i])
        data.author = authors
        if (self.request.get('abstract') != None):
            data.abstract =
urllib.unquote(self.request.get('abstract'))
        else:
            data.abstract = None
        if (self.request.get('url').strip()):
            data.url = self.request.get('url').strip()
        else:
            data.url = None
        if (self.request.get('address').strip()):
            data.address =
self.request.get('address').strip()

        data.put()
        output += 'Thank you for editing. <a
href="/DocumentList">Back to list</a>'
        if error != None:
            output += 'There was an error: '+error

        template_values = {
            'bodytext': output,
            'logintext': logintext,
        }

        path = os.path.join(os.path.dirname(__file__),
'templates/generic.html')
        self.response.out.write(template.render(path,
template_values))

class autosuggestHandler(webapp.RequestHandler):
    def get(self):
        output = ''
        term = self.request.get('q')
        fieldname = self.request.get('fieldname')
        #query = Person.gql("WHERE lastname = :1",
self.request.get('q'))
        query = Person.gql("WHERE lastname >= :1 AND lastname
< :2 LIMIT 8", term, term + u"\ufffd")
        count = 0
        output += '<ul>'
        for person in query:
            name = person.lastname+', '+person.firstname
            if (person.middlename):
                name += ' '+person.middlename
            if (fieldname == 'advisorlist'):

```

```

        output += '<li><a
href="javascript:autosuggest_addItem_advisor(\''+str(person.key(
))+'\', \''+name+'\');">'+name
        else:
            output += '<li><a
href="javascript:pickFromList(\''+str(person.key())+'\','
\''+name+'\',' \''+fieldname+'\');">'+name
            query2 = Affiliation.gql("WHERE person = :1",
person)
            institcount = 0
            instits = ''
            for affil in query2:
                if (institcount >= 1):
                    instits += ', '
                    instits += affil.institution.name + ' - ' +
affil.affiliation
                if (instits):
                    output +=
'<small>'+instits+'</small></a></li>'
                    count = count + 1
            output += '</ul>'
            if (count < 1):
                output = '<font size="1">No results found. The
search is case sensitive. You can also add a new person.</font>'
            self.response.out.write(output)

```

```

class MainPage(webapp.RequestHandler):
    def get(self, directory):
        if directory is None or directory == "":
            directory = defaultRedirect
            self.redirect("/") + directory + "/"

application = webapp.WSGIApplication(
    [
        ("/Institution",InstitutionFormHandler),
        ("/InstitutionList",InstitutionListHandler),
        ("/InstitutionEdit",InstitutionEditHandler),
        ("/Person",PersonFormHandler),
        ("/PersonList",PersonListHandler),
        ("/PersonEdit",PersonEditHandler),
        ("/Thesis",ThesisFormHandler),
        ("/ThesisList",ThesisListHandler),
        ("/ThesisEdit",ThesisEditHandler),
        ("/Affiliation",AffiliationFormHandler),
        ("/AffiliationList",AffiliationListHandler),
        ("/DocumentList",DocumentListHandler),

```

```

        ("/DocumentEdit",DocumentEditHandler),
        ("/insert",InsertHandler),
        ("/autosuggest",autosuggestHandler),
        ("/getinfo",getInfoHandler)
        #(r'/(.*)', MainPage)
    ],
    debug=True)

def main():
    global user
    global logintext
    global template_values
    user = users.get_current_user()
    if user:
        logintext = 'You are logged in as %s (<a
href="%s">sign out</a>)' % (user.nickname(),
users.create_logout_url('/'))
    else:
        logintext = 'Please <a href="%s">sign in</a> to get
credit for your work.' % users.create_login_url('/')
        logintext += '<br /><br />'
        run_wsgi_app(application)

if __name__ == "__main__":
    main()

```

templates/thesis.html

```

{% extends "base.html" %}

{% block sidebar %}
<h3>Thesis Links</h3>


- <a
href="javascript:showdiv('addinstitution');">[Add
Institution]</a></li>
- <a
href="javascript:showdiv('addperson');">[Add Person]</a></li>


<h3>Other Instructions</h3>
<div class="content">
<font size='2'>Other documents can either be entered
individually, or in

```

```

                <a href="http://en.wikipedia.org/wiki/BibTeX"
target="_blank">BibTeX</a> format.</font>
</div>

{% endblock %}

{% block content %}
<script src="/js/BibTeX.js"></script>

<script type="text/javascript"
src="/js/autosuggest_list.js"></script>
<script type="text/javascript" src="/js/doclist.js"></script>
<script type="text/javascript"
src="/js/advisorlist.js"></script>
<script type="text/javascript"
src="/js/docauthorlist.js"></script>
<link href="/js/doclist.css" rel="stylesheet" type="text/css" />

<script type="text/javascript" src="/js/sendbibtex.js"></script>

<script language="Javascript">
// this will check the thesis form to be sure all is filled out
before it is submitted

function checkform()
{
if (document.thesis.institution.value == "") {
    alert('You must enter an institution.');
```

```

    }
    if (document.thesis.year.value == "") {
        alert('You must enter the publication year.');
```

```

        return false;
    }
    if (document.thesis.url.value == "") {
        alert('You must enter a URL. If there is absolutely no
URL available, just type n/a');
```

```

        return false;
    }
    if (document.thesis.doctype.value == "") {
        alert('You must enter the document type.');
```

```

        return true;
    }
</script>
```

```
<style type="text/css">
```

```
.box
```

```
{
```

```

    width: 700px;
    height: 400px;
    margin: auto;
    background-color: #F0F0F0;
    position: absolute;
    border: 1px solid black;
    padding: 25px 10px 10px 10px;
    z-index: 10;
    top: 50%;
    left: 50%;
    margin-top: -100px;
    margin-left: -350px;
    overflow: auto;
```

```
}
```

```
</style>
```

```
<script language="Javascript">
```

```
function delFields(form)
```

```
{
```

```
    for (i=0; i<form.elements.length; i++)
```

```
    {
```

```
        if (form.elements[i].type != 'button' &&
form.elements[i].type != 'submit')
```

```
        {
```

```

        form.elements[i].value = '';
    }
}

function hidediv(div)
{
    document.getElementById(div).style.display = "none";
    for (n=0; n<document.forms.length; n++)
    {
        if (document.forms[n].parentNode.id == div)
delFields(document.forms[n]);
    }
    if (div == 'adddoc')
    {
        deleteAll_docauthor();
        //itemlist_advisor = {"array":[]};
        createTable_docauthor();
        //alert('got it');
    }
}

function showdeadcenterdiv(Xwidth,Yheight,divid) {
// First, determine how much the visitor has scrolled

    var scrolledX, scrolledY;
    if( self.pageYoffset ) {
        scrolledX = self.pageXoffset;
        scrolledY = self.pageYoffset;
    } else if( document.documentElement &&
document.documentElement.scrollTop ) {
        scrolledX = document.documentElement.scrollLeft;
        scrolledY = document.documentElement.scrollTop;
    } else if( document.body ) {
        scrolledX = document.body.scrollLeft;
        scrolledY = document.body.scrollTop;
    }

    // Next, determine the coordinates of the center of
browser's window

    var centerX, centerY;
    if( self.innerHeight ) {
        centerX = self.innerWidth;
        centerY = self.innerHeight;
    } else if( document.documentElement &&
document.documentElement.clientHeight ) {

```

```

        centerX = document.documentElement.clientWidth;
        centerY = document.documentElement.clientHeight;
    } else if( document.body ) {
        centerX = document.body.clientWidth;
        centerY = document.body.clientHeight;
    }

    // Xwidth is the width of the div, Yheight is the height of
the
    // div passed as arguments to the function:
    var leftoffset = scrolledX + (centerX - Xwidth) / 2;
    var topoffset = scrolledY + (centerY - Yheight) / 2;
    // The initial width and height of the div can be set in
the
    // style sheet with display:none; divid is passed as an
argument to // the function
    var o=document.getElementById(divid);
    var r=o.style;
    r.position='absolute';
    r.top = topoffset + 50 + 'px';
    r.left = leftoffset + 350 + 'px';
    r.display = "block";
}

function showdiv(div)
{
    if (div != 'explaindocs')
        showdeadcenterdiv('700','400',div)
    else
        document.getElementById(div).style.display = "block";
}
</script>

<div class="box" id="adddoc_bibtex" style="display: none;">
<b>Enter a new Document</b>
    <form onsubmit="return false;" name="adddoc_bibtex">
    <table border="0">
    <tr><td>Bibtex:</td><td><textarea name="bibtex"
rows="4" cols="40"></textarea></td></tr>
    <tr><td>&nbsp;</td><td><input type="button"
value="Submit" onclick="insertDoc_bibtex();"></td></tr>
    </table>
    </form>
    <div align="right"><a
href="javascript:hidediv('adddoc_bibtex');">[Close]</a></div><br
/>
</div>

```

```

<div class="box" id="adddoc_prev" style="display: none;">
<b>Enter a new Document</b>
  <form onsubmit="return false;" name="adddoc_prev">
    <table border="0">
      <tr><td>Document:</td><td><select name="doc"><option
value="">Select a Document</option>
      {% for mydoc in list_docs %}
        <option value="{{ mydoc.key }}">{{
mydoc.title }}</option>
      {% endfor %}
    </select></td></tr>
    <tr><td>&nbsp;</td><td><input type="button"
value="Submit" onclick="insertDoc_prev();"></td></tr>
    </table>
  </form>
  <div align="right"><a
href="javascript:hidediv('adddoc_prev');">[Close]</a></div><br
/>
</div>

```

```

<div class="box" id="other_advisors" style="display: none;">
<b>Enter an Advisor</b>
  <form onsubmit="return false;" name="other_advisors">
    <table border="0">
      <tr><td>Advisor:
      <a href="javascript:showdiv('addperson');"
onClick="curpers='';"></a>
    </td><td><div id='docadvisorlist'><select
name="author"><option value="">Select Person</option>
      {% for myperson in list_person %}
        <option value="{{ myperson.key }}">{{
myperson.lastname }}, {{ myperson.firstname }}{% if
myperson.middlename %} {{ myperson.middlename }} {% endif
%}</option>
      {% endfor %}
    </select></div></td></tr>
    <tr><td>&nbsp;</td><td><input type="button"
value="Submit" onclick="insertAdvisor();"></td></tr>
    </table>
  </form>
  <div align="right"><a
href="javascript:hidediv('other_advisors');">[Close]</a></div><b
r />
</div>

```

```

<div class="box" id="adddoc" style="display: none;">
<b>Enter a new Document</b>
  <form onsubmit="return false;" name="adddoc">
    <table border="0">
      <tr><td>Title*:</td><td><input type="text"
name="title" size="40"></td></tr>
      <tr>
        <td>Authors*:</td>
        <td>
          <div id='docauthorlist'><select
name="author"><option value="">Select Person</option>
          {% for myperson in list_person %}
            <option value="{{ myperson.key }}">{{
myperson.lastname }}, {{ myperson.firstname }}{% if
myperson.middlename %} {{ myperson.middlename }} {% endif
%}</option>
          {% endfor %}
        </select> <input type="button" value="Add Author to
Document" onClick="insertAuthor();"></div>
          <div id='docauthormultilist'></div>
        </td>
      </tr>
      <tr><td>Abstract:</td><td><textarea name="abstract"
rows="4" cols="40"></textarea></td></tr>
      <tr><td>URL:</td><td><input type="text" name="url"
size="40"></td></tr>
      <tr><td>Year Published*:</td><td><select
name="year"><option value="">Select a Year</option>
          {% for i in list_years %}
            <option value="{{ i }}">{{ i }}</option>
          {% endfor %}
        </select></td></tr>
      <tr><td>&nbsp;</td><td><input type="button"
value="Submit" onclick="insertDoc();"></td></tr>
    </table>
    <textarea name="bibtex" style="display:
none;"></textarea>
  </form>
  <div align="right"><a
href="javascript:hidediv('adddoc');">[Close]</a></div><br />
</div>

```

```

<div class="box" id="addinstitution" style="display: none;">
<b>Enter a new Institution</b>
  <form onsubmit="return false;" name="addinstitution">
    <table border="0">

```

```

        <tr><td>Name*:</td><td><input type="text" name="name"
size="40"></td></tr>
        <tr><td>City:</td><td><input type="text" name="city"
size="40"></td></tr>
        <tr><td>State*:</td><td><input type="text"
name="state" size="40"></td></tr>
        <tr><td>Zip:</td><td><input type="text" name="zip"
size="40"></td></tr>
        <tr><td>Country:</td><td><input type="text"
name="country" size="40"></td></tr>
        <tr><td>&nbsp;</td><td><input type="button"
value="Submit" onclick="insertInstitution();"></td></tr>
    </table>
</form>
    <div align="right"><a
href="javascript:hidediv('addinstitution');">[Close]</a></div><b
r />
</div>

```

```

<div class="box" id="addperson" style="display: none;">
<b>Enter a new Person</b>
    <form onsubmit="return false;" name="addperson">
    <table border="0">
    <tr><td colspan="2"><b>Identification</b></td></tr>
    <tr><td>First Name*:</td><td><input type="text"
name="firstname" size="40"></td></tr>
    <tr><td>Middle Name:</td><td><input type="text"
name="middlename" size="40"></td></tr>
    <tr><td>Last Name*:</td><td><input type="text"
name="lastname" size="40"></td></tr>
    <tr><td>E-mail Address:</td><td><input type="text"
name="email" size="40"></td></tr>
    <tr><td colspan="2"><b>Affiliation</b></td></tr>
    <tr><td>Institution:</td><td><div id='persinstitlist'>
    <select name="institution"><option value="">Select
Institution</option>
    {% for myinstit in list_institution %}
        <option value="{{ myinstit.key }}">{{
myinstit.name }}</option>
    {% endfor %}
    </select> </div></td></tr>
    <tr><td>Affiliation:</td><td><select
name="affiliation"><option value="">Select an
Affiliation</option>
    {% for type in a_types %}
        <option value="{{ type }}">{{ type }}</option>
    {% endfor %}

```

```

        </select></td></tr>
        <tr><td>&nbsp;</td><td><input type="button"
value="Submit" onclick="insertPerson();"></td></tr>
    </table>
</form>
<br />
    <div align="right"><a
href="javascript:hidediv('addperson');">[Close]</a></div><br />
</div>

<script type="text/javascript" language="javascript">

function showWait()
{
    mydiv = document.getElementById('waitimg');
    mydiv.style.display='block';
}
function hideWait()
{
    mydiv = document.getElementById('waitimg');
    mydiv.style.display='none';
}
function getReq()
{
    var req = false;
    if (window.XMLHttpRequest) {
        req = new XMLHttpRequest();
    } else if (window.ActiveXObject) {
        req = new ActiveXObject("Microsoft.XMLHTTP");
    } else {
        alert("Your browser does not support AJAX.");
        return false;
    }
    return req;
}

function insertPerson() {
    showWait();
    var req = false;
    req = getReq();

    req.onreadystatechange = function() {
        if (req.readyState == 4 && req.status == 200) {
            var rtext = eval('(' + req.responseText + ')');
            if (rtext.result == 'success') {

```



```

    //document.getElementById('advisorlist').innerHTML =
    '<select name="advisor"><option value="">Select
    Person</option>'+response+'</select> ';
    //document.getElementById('authorlist').innerHTML
    = '<select name="author"><option value="">Select
    Person</option>'+response+'</select> ';

```

```

    document.getElementById('docauthorlist').innerHTML =
    '<select name="author"><option value="">Select
    Person</option>'+response+'</select> <input type="button"
    value="Add Author to Document" onClick="insertAuthor();">';

```

```

    document.getElementById('docadvisorlist').innerHTML =
    '<select name="author"><option value="">Select
    Person</option>'+response+'</select> ';

```

```

    //document.forms["thesis"].elements["advisor"].value =
    oldval_advisor;

```

```

    //document.forms["thesis"].elements["author"].value =
    oldval_author;

```

```

        if (key && curpers != '') {

```

```

            //document.forms["thesis"].elements[curpers].value = key;
            pickFromList(key, name, curpers);
            curpers = '';

```

```

        }

```

```

        hidediv('addperson');
        hideWait();

```

```

    }

```

```

}
req.open('GET', '/getinfo?type=person', true)
req.send(null)

```

```

}

```

```

function insertInstitution() {

```

```

    showWait();
    var req = false;
    req = getReq();

```

```

    req.onreadystatechange = function() {
        if (req.readyState == 4 && req.status == 200) {
            var rtext = eval('(' + req.responseText + ')');

```

```

        if (rtext.result == 'success')
            updateInstitution(rtext.key);
        else {
            alert(rtext.error);
            hideWait();
        }
    }
}
name =
encodeURIComponent(document.forms["addinstitution"].elements["name"].value);
city =
encodeURIComponent(document.forms["addinstitution"].elements["city"].value);
state =
encodeURIComponent(document.forms["addinstitution"].elements["state"].value);
zip =
encodeURIComponent(document.forms["addinstitution"].elements["zip"].value);
country =
encodeURIComponent(document.forms["addinstitution"].elements["country"].value);
req.open('GET',
'/insert?type=institution&name='+name+'&city='+city+'&state='+state+'&zip='+zip+'&country='+country, true)
req.send(null)
}

extravar = 0;
function updateInstitution(newval) {
    req = getReq();
    oldvalue =
document.forms["thesis"].elements["institution"].value;
    req.onreadystatechange = function() {
        if (req.readyState == 4 && req.status == 200) {
            var response = req.responseText;
            var link = '<a
href="javascript:showdiv(\'addinstitution\');">[Add
Institution]</a>';

            document.getElementById('institutionlist').innerHTML =
'<select name="institution" id="institution_updated"><option
value="">Select Institution</option>'+response+'</select> ';

            document.getElementById('persinstitlist').innerHTML =

```

```

'<select name="institution"><option value="">Select
Institution</option>'+response+'</select> ';

        if (newval && extravar==1) {

            document.forms["thesis"].elements["institution"].value =
newval;
                extravar = 0;
            }
            else {

                document.forms["thesis"].elements["institution"].value =
oldvalue;
                    }
                    hiddenDiv('addinstitution');
                    hideWait();
                }
            }
            req.open('GET', '/getinfo?type=institution', true)
            req.send(null)
        }

function insertDoc() {
    showWait();
    var req = false;
    req = getReq();

    req.onreadystatechange = function() {
        if (req.readyState == 4 && req.status == 200) {
            updateDoc(req.responseText);
        }
    }
    title =
encodeURIComponent(document.forms["adddoc"].elements["title"].value);
    author =
encodeURIComponent(document.forms["adddoc"].elements["key_docauthor"].value);
    abstract =
encodeURIComponent(document.forms["adddoc"].elements["abstract"].value);
    url =
encodeURIComponent(document.forms["adddoc"].elements["url"].value);
    year =
encodeURIComponent(document.forms["adddoc"].elements["year"].value);
    bibtex2 =
escape(encodeURIComponent(document.forms["adddoc"].elements["bibtex"].value));
}

```

```

        var parameters =
        'bibtex='+bibtex2+'&type=doc&title='+title+'&author='+author+'&a
bstract='+abstract+'&url='+url+'&year='+year;
        //req.open('GET',
        '/insert?type=doc&title='+title+'&author='+author+'&abstract='+a
bstract+'&url='+url+'&year='+year, true)
        //req.send(null)

        req.open('POST', '/insert', true);
        req.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
        req.setRequestHeader("Content-length", parameters.length);
        req.setRequestHeader("Connection", "close");
        req.send(parameters);
    }

function updateDoc(doc) {
    var rtext = eval('(' + doc + ')');
    //document.getElementById('testblock').innerHTML =
response.wclass[0].name;
    //document.getElementById('citations').innerHTML =
rtext.author;
    if (rtext.result == 'success') {
        addItem(doctype, rtext.key, rtext.title,
rtext.author);
        hidediv('adddoc');
    }
    else
        alert(rtext.error);
    hideWait();
}

function insertDoc_prev() {
    showWait();
    if (document.forms["adddoc_prev"].elements["doc"].value ==
"")
        return;
    var req = false;
    req = getReq();

    req.onreadystatechange = function() {
        if (req.readyState == 4 && req.status == 200) {
            updateDoc_prev(req.responseText);
        }
    }
}

```

```

        doc =
encodeURIComponent(document.forms["adddoc_prev"].elements["doc"].value);
        var parameters = 'type=doc_prev&doc='+doc;
        req.open('GET', '/insert?'+parameters, true)
        req.send(null)
    }

function updateDoc_prev(doc) {
    var rtext = eval('(' + doc + ')');
    addItem(doctype, rtext.key, rtext.title, rtext.author);
    hiddenDiv('adddoc_prev');
    hideWait();
}

```

```

</script>
<script type="text/javascript"
src="/js/autosuggest.js"></script>

```

```

<style type="text/css">
/* ----- */
/* CUSTOMIZE AUTOSUGGEST STYLE */
.autosuggest_results{width:260px; border:solid 1px #000;
display:none; background-color: #DFDFDF;}
.autosuggest_results ul, #results li{padding:0; margin:0;
border:0; list-style:none;}
.autosuggest_results li {border-top:solid 1px #DEDEDE;}
.autosuggest_results li a{display:block; padding:4px; text-
decoration:none; color:#000000; font-weight:bold;}
.autosuggest_results li a small{display:block; text-
decoration:none; color:#999999; font-weight:normal;}
.autosuggest_results li a:hover{background:#FFFFCC;}
.autosuggest_results ul {padding:6px;}

.autosuggest_picklist a {background-color: #DFECCF; padding: 5 5
5 5; margin: 5 5 5 5;}
.autosuggest_picklist a:hover {background-color: #D1DDEF; }
</style>

```

```

<script language="javascript">
function finishPicklist(fieldname)
{
    e =
document.getElementById('autosuggest_results_'+fieldname);

```

```

        e.style.display='none';
        document.getElementById('autosuggest_search_'+fieldname).value='';
    }
    function pickFromList(id, name, fieldname)
    {
        div =
        document.getElementById('autosuggest_picklist_'+fieldname);
        div.innerHTML = '<a
href="javascript:remPerson(\''+fieldname+\');">'+name+'</a>';
        div.innerHTML += '<input type="hidden" name="'+fieldname+'
value="'+id+'" />';
        finishPicklist(fieldname);
    }
    function remPerson(fieldname)
    {
        div =
        document.getElementById('autosuggest_picklist_'+fieldname);
        div.innerHTML='';
    }

    autosuggest_lastone = ''
    function autosuggest_useme(fieldname)
    {
        if
        (document.getElementById('autosuggest_search_'+fieldname).value=
        '' && autosuggest_lastone != fieldname)
        {
            e =
            document.getElementById('autosuggest_results_'+fieldname);
            e.innerHTML = 'Start typing the last name. The search
            is case sensitive.';
            e.style.display='block';
            autosuggest_lastone = fieldname;
        }
    }
</script>

<h2>Enter a Thesis</h2>
    <form name="thesis" method="POST" action="{ { post_to
}} " onSubmit="return checkform()" width="100%">
    <input type="hidden" name="type" value="thesis" />
    { { otherhiddenfields } }
    If you need to add an institution or a person (any
    author or advisor), please use the links to the left.
    <br /><br />

```

```

    {{ logmein }}
    <table border="0">
    <tr><td>Institution*:
        <a href="javascript:showdiv('addinstitution');"
onClick="extravar=1;"></a>
    </td><td><div id="institutionlist">
        <select name="institution"><option value="">Select
Institution</option>
        {% for myinstit in list_institution %}
            <option value="{{ myinstit.key }}" {% if
doc_institution %}{% ifequal myinstit.key doc_institution %}
selected{% endifequal %}{% endif %}>{{ myinstit.name }}</option>
        {% endfor %}
    </select> </div></td></tr>
    <tr><td>Title*:</td><td><input type="text"
name="title" size="40" value="{{ doc_title }}">
    </td></tr>
    <tr><td>Author*: <a
href="javascript:showdiv('addperson');"
onClick="curpers='author';"></a>
    <br />
    <a href="#" onClick="alert('To add an author, click on
the plus sign or the Add a Person on the left sidebar. Remember,
the author search is by last name and case sensitive.');"
style="font-size: small;">author not there?</a>

    </td><td>
    <input id="autosuggest_search_author" type="text"
onkeyup="javascript:autosuggest('author')"
onfocus="autosuggest_useme('author');"/>
    <div id="autosuggest_results_author"
class="autosuggest_results" style="position: absolute;"></div>
    <div id="autosuggest_picklist_author"
class="autosuggest_picklist"></div>
    </td></tr>
    <tr><td>Abstract:</td><td><textarea name="abstract"
rows="4" cols="60">{{ doc_abstract }}</textarea></td></tr>
    <tr><td>Keywords:</td><td><input type="text"
name="keywords" size="40" value="{{ doc_keywords }}"></td></tr>
    <tr><td>Main Advisor*: <a
href="javascript:showdiv('addperson');"
onClick="curpers='advisor';"></a></td>

```

```
 <input id="autosuggest_search_advisor" type="text" onkeyup="javascript:autosuggest('advisor') " onfocus="autosuggest_useme('advisor');"/>    140  المنارة للاستشارات  www.manaraa.com |
```

(title, abstract, etc.). Finally, if the document is already in the system, you can

just add it that way.

```
</div>
</td></tr>
<tr><td>Citations:<br />
</td>
<td>
```

New: Add BibTex ref | Add Document OR Reference Previous Document

```
<div id='citations'></div>
```

```
</td>
```

```
</tr>
```

```
<tr><td>References:</td>
```

```
<td>
```

New: Add BibTex ref | Add Document OR Reference Previous Document

```
<div id='references'></div>
```

```
</td>
```

```
</tr>
```

```
<tr><td>Resulting Publications:</td>
```

```
<td>
```

New: Add BibTex ref | Add Document OR Reference Previous Document

```
<div id='publications'></div>
```

```
</td>
```

```
</tr>
```

```
<tr><td>&nbsp;</td><td><input type="submit"></td></tr>
```

```
</table>
```

```
</form>
```

```
<br /><br />
```

* Required field

```
        {{ output }}  
{% endblock %}
```

APPENDIX D. BROWSE AND SEARCH PAGE CODE

Import.php

```
<?
$url = 'http://localhost:8080/backup/';
$url = 'http://it-thesis.appspot.com/backup/';
$delim = ' ### ';
$newline = "###\n\n###";

require_once("includes/config.inc.php");
require_once("includes/dbconnect.php");

$sqlbase = dbConnect();

mysql_query("TRUNCATE person"); echo mysql_error();
$file = file_get_contents($url.'person');
$cut = explode($newline, $file);
foreach ($cut as $eachline)
{
    $line = explode($delim, $eachline);
    if (sizeof($line) >= 2) {
        $sql = "INSERT INTO person SET ".

        "personID='".mysql_real_escape_string($line[0])."', ".

        "lastname='".mysql_real_escape_string($line[1])."', ".

        "firstname='".mysql_real_escape_string($line[2])."', ".

        "middlename='".mysql_real_escape_string($line[3])."', ".
        "email='".mysql_real_escape_string($line[4])."',
        ".

        "user_add='".mysql_real_escape_string($line[5])."', ".

        "user_modify='".mysql_real_escape_string($line[6])."' ";
        mysql_query($sql); echo mysql_error();
        #echo $sql;
    }
}
```

```

}

mysql_query("TRUNCATE institution"); echo mysql_error();
$file = file_get_contents($url.'institution');
$cut = explode($newline, $file);
foreach ($cut as $eachline)
{
    $line = explode($delim, $eachline);
    if (sizeof($line) >= 2) {
        $sql = "INSERT INTO institution SET ".

        "institutionID='".mysql_real_escape_string($line[0])."', ".
        "name='".mysql_real_escape_string($line[1])."',
        ".
        "city='".mysql_real_escape_string($line[2])."',
        ".
        "state='".mysql_real_escape_string($line[3])."',
        ".
        "zip='".mysql_real_escape_string($line[4])."', ".

        "country='".mysql_real_escape_string($line[5])."', ".

        "user_add='".mysql_real_escape_string($line[6])."', ".

        "user_modify='".mysql_real_escape_string($line[7])."' ";
        mysql_query($sql); echo mysql_error();
        #echo $sql;
    }
}

```

```

mysql_query("TRUNCATE affiliation"); echo mysql_error();
$file = file_get_contents($url.'affiliation');
$cut = explode($newline, $file);
foreach ($cut as $eachline)
{
    $line = explode($delim, $eachline);
    if (sizeof($line) >= 2) {
        $sql = "INSERT INTO affiliation SET ".

        "affiliationID='".mysql_real_escape_string($line[0])."', ".

        "institutionID='".mysql_real_escape_string($line[1])."', ".

        "personID='".mysql_real_escape_string($line[2])."', ".
    }
}

```

```

"affiliation='".mysql_real_escape_string($line[3])."', ".
"user_add='".mysql_real_escape_string($line[4])."', ".

"user_modify='".mysql_real_escape_string($line[5])."' ";
mysql_query($sql); echo mysql_error();
#echo $sql;
}
}

mysql_query("TRUNCATE genericdoc"); echo mysql_error();
$file = file_get_contents($url.'document');
$cut = explode($newline, $file);
foreach ($cut as $eachline)
{
    $line = explode($delim, $eachline);
    if (sizeof($line) >= 2) {
        $sql = "INSERT INTO genericdoc SET ".

        "genericdocID='".mysql_real_escape_string($line[0])."', ".
            "title='".mysql_real_escape_string($line[1])."',
".

        "abstract='".mysql_real_escape_string($line[2])."', ".
            "url='".mysql_real_escape_string($line[3])."', ".

        "pubyear='".mysql_real_escape_string($line[4])."', ".

        "keywords='".mysql_real_escape_string($line[5])."', ".
            "author='".mysql_real_escape_string($line[6])."',
".
            "bibtex='".mysql_real_escape_string($line[7])."',
".

        "user_add='".mysql_real_escape_string($line[8])."', ".

        "user_modify='".mysql_real_escape_string($line[9])."', ".

        "user_nick='".mysql_real_escape_string($line[10])."', ".

        "user_nick_mod='".mysql_real_escape_string($line[11])."',
".

        "modifydate='".mysql_real_escape_string($line[12])."' ";
mysql_query($sql); echo mysql_error();

```

```

    }
}

mysql_query("TRUNCATE thesis"); echo mysql_error();
$file = file_get_contents($url.'thesis');
$cut = explode($newline, $file);
foreach ($cut as $eachline)
{
    $line = explode($delim, $eachline);
    if (sizeof($line) >= 2) {
        $sql = "INSERT INTO thesis SET ".

        "thesisID='".mysql_real_escape_string($line[0])."', ".
        "title='".mysql_real_escape_string($line[1])."',
        ".

        "abstract='".mysql_real_escape_string($line[2])."', ".
        "url='".mysql_real_escape_string($line[3])."', ".

        "pubyear='".mysql_real_escape_string($line[4])."', ".

        "keywords='".mysql_real_escape_string($line[5])."', ".
        "author='".mysql_real_escape_string($line[6])."',
        ".

        "doctype='".mysql_real_escape_string($line[7])."', ".

        "institutionID='".mysql_real_escape_string($line[8])."', ".

        "advisor='".mysql_real_escape_string($line[9])."', ".

        "otheradvisors='".mysql_real_escape_string($line[10])."',
        ".

        "pubs_citations='".mysql_real_escape_string($line[11])."',
        ".

        "pubs_refs='".mysql_real_escape_string($line[12])."', ".

        "pubs_resulting='".mysql_real_escape_string($line[13])."',
        ".

        "user_add='".mysql_real_escape_string($line[14])."', ".

        "user_modify='".mysql_real_escape_string($line[15])."', ".
    }
}

```

```

        "user_nick='".mysql_real_escape_string($line[16])."', ".
        "user_nick_mod='".mysql_real_escape_string($line[17])."',
    ".

        "modifydate='".mysql_real_escape_string($line[18])."' ";
        mysql_query($sql); echo mysql_error();

    }
}

// make list of citations to theses
mysql_query("TRUNCATE citations_to_thesis"); echo mysql_error();
$sql = "SELECT thesisID,pubs_citations FROM thesis";
$query = mysql_query($sql); echo mysql_error();
while ($row = mysql_fetch_array($query))
{
    $cits = explode(", ", $row[1]);
    foreach ($cits as $value)
    {
        $sql2 = "INSERT INTO citations_to_thesis SET
        genericdocID='".mysql_real_escape_string($value)."',
        thesisID='".mysql_real_escape_string($row[0])."' ";
        if ($value)
        {
            mysql_query($sql2); echo mysql_error();
        }
    }
}
echo "done.";
dbClose($sqlbase);
?>

```

browse.php

```

<?
include_once("includes/overall_header.inc.php");
require_once("includes/config.inc.php");
require_once("includes/dbconnect.php");
$sqlbase = dbConnect();

```


search.php

```
<?
include_once("includes/overall_header.inc.php");
require_once("includes/config.inc.php");
require_once("includes/dbconnect.php");
$sqlbase = dbConnect();

$action = $_REQUEST[action];
if (empty($action)) $action = "showform";

?>
    <form method="GET" action="<?= $_SERVER[PHP_SELF]; ?>">
        <input type="hidden" name="action" value="search">

        <select name="type">
            <option>All</option>
            <option>Title</option>
            <option>Abstract</option>
            <option>Keywords</option>
        </select>
        <input type="text" size="40" name="q" value="<?=
htmlentities($_REQUEST[q]); ?>">
        <input type="submit" value="search">
    </form>
<br /><br />
    <?
if ($action == "search")
{
    $type = $_REQUEST[type];
    $q = mysql_real_escape_string($_REQUEST[q]);
    if ($type == "All")
    {
        $where = "(title LIKE \"%$q%\" OR abstract LIKE
\\\"$q%\" OR keywords LIKE \"%$q%\")";
    }
    elseif ($type == "Title")
    {
        $where = "title LIKE \"%$q%\"";
    }
    elseif ($type == "Abstract")
    {
        $where = "abstract LIKE \"%$q%\"";
    }
    elseif ($type == "Keywords")
```

```

    {
        $where = "keywords LIKE \"%$q%\"";
    }
    $sql = "SELECT t.*, CONCAT(p.firstname, ' ', p.middlename,
' ', p.lastname) as theauthor, i.name as institution FROM thesis
as t ".
        "LEFT JOIN person as p ON t.author =
p.personID ".
        "LEFT JOIN institution as i ON
t.institutionID = i.institutionID ".
        "WHERE $where ".
        "ORDER BY title";
    $query = mysql_query($sql); echo mysql_error();
    $numrows = mysql_num_rows($query);
    ?>
    Listing a total of <i><?= $numrows; ?></i> rows<br /><br />
    <link rel="stylesheet" type="text/css"
href="linkstyle.css">
    <table border="0" cellpadding="1" cellspacing="1"
width="100%" border='1'>
        <tr class="sp_title">
            <td colspan="14">Theses</td>
        </tr>
        <tr class="sp_columns">
            <td>&nbsp;</td>
            <td>Title</td>
            <td>Institution</td>
            <td>Year</td>
            <td>Author</td>
            <td>Type</td>
        </tr>
        <?
        while ($row = mysql_fetch_array($query))
        {
            ?>
            <tr class="sp_data">
                <td><a href="thesis.php?id=<?=
$row[thesisID]; ?>"><img src='images/mglass.gif'
border='0'></a></td>
                    <td><?= $row[title]; ?></td>
                    <td><?= $row[institution]; ?></td>
                    <td><?= $row[pubyear]; ?></td>
                    <td><?= $row[theauthor]; ?></td>
                    <td><?= $row[doctype]; ?></td>
            </tr>
            <?
        }

```

```
        echo "</table>";  
    }  
    dbClose($sqlbase);  
    include_once("includes/overall_footer.inc.php");  
?>
```