

# Dancing with the Devil: Faculty Assessment Process Transformed with Web Technology

Luis O. Hernández  
Institutional Technology  
Ringling School of Art & Design  
Sarasota, FL 34234  
941-359-7633

lhernand@ringling.edu

Karen Wetherby  
Dean of Faculty  
Ringling School of Art & Design  
Sarasota, FL 34234  
941-359-7536

kwetherb@ringling.edu

Mahmoud Pegah  
Institutional Technology  
Ringling School of Art & Design  
Sarasota, FL 34234  
941-359-7633

mpegah@ringling.edu

## ABSTRACT

Ringling School's Institutional Technology, in partnership with the Dean of Faculty Office has assessed, planned, and implemented a web-based paperless Course and Faculty Assessment package using home-brewed software components. At our institution, the Course and Faculty Assessment package has transformed the traditional faculty evaluation process into a pedagogically effective paperless protocol. Although the faculty assessment process is straight forward, it is a labor-intensive activity for the Academic Affairs division. Furthermore, managing and organizing the assessment process requires coordination with faculty members. Our web-based approach to course and faculty assessment is a cost effective, robust, scalable, flexible, adaptable, secure, and attractive solution.

In this report, we present key topics including project strategic planning for Institutional Technology, implementation framework, archiving and accessing past assessments, data analysis issues, user support issues, campus Institutional Technology standards, integration plans with a campus portal, and other future plans. In light of our experience at Ringling School, we discuss issues we had to address relating to building and maintaining pedagogically effective assessment tools. We discuss our method and present our approach to evaluate our success in delivering this strategic change.

## Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services – *Data sharing, Web-based services.*

## General Terms

Management, Measurement, Performance.

## Keywords

Apache, Course Evaluation, Database, Development, Faculty, History, Java, LDAP, MySQL, Testing, Tomcat

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*SIGUCCS'04*, October 10–13, 2004, Baltimore, Maryland, USA.  
Copyright 2004 ACM 1-58113-869-5/04/0010...\$5.00.

## 1. INTRODUCTION

Faculty use course and faculty evaluations as a major piece of information that helps them improve their courses and instruction [3]. The traditional method involves distribution of evaluation materials, collection, tally, and distribution of results. While the traditional approach is effective, it is a labor-intensive activity for staff within the Academic Affairs division and also uses a considerable amount of paper. A new approach that involves electronic, anonymous evaluations eliminates the dependency on paper, allows for students to fill out the evaluations accurately with a consistent interface, allows for faculty to view tabulated evaluation results for their courses, and allows for the Academic Affairs division to concentrate on the contents of the evaluation as opposed to the preparations for evaluation distribution and collection of results.

While many institutions still use the traditional approach of paper and pencil to evaluate courses and faculty, numerous institutions of higher education have migrated to a digital solution [14]. In this report, we will discuss the online Course and Faculty Assessment (CFA) package developed at our institution: we will share our framework and experience during development of a home-brewed system; we will discuss key issues related with early involvement of campus stakeholders and user support; and we will also discuss how the CFA package can be used at other institutions to archive assessment data and leverage the power of a database to easily obtain assessment information.

## 2. THE NEED FOR CHANGE

Traditional methods for course and faculty evaluations involve a four-step process, which the Dean of Faculty (DOF) Office oversees. The DOF Office distributes evaluations to the faculty, then collects them, makes copies of all evaluations results, and finally distributes the completed evaluations back to the faculty who use the results to improve their courses.

### 2.1 Distribution of Evaluations

Academic Affairs approves changes to the course evaluation form after recommendations from faculty committees. After approval of the evaluation, staff members in the DOF Office make copies of the evaluation form for every student in every course. The amount of copies for a small institution translates to about five thousand pages for one semester. Each academic year, the total amount is ten thousand. When results are collected, the completed evaluations are copied again for each recipient so that the DOF Office may archive a copy, which could double the amount to twenty thousand pages per academic year. Some of the evaluations will be useful for faculty to improve courses, but

depending on the evaluation and the evaluator, some evaluations may be useless to the faculty, making those evaluations paper waste.

## **2.2 Collection of Evaluations**

Collection of evaluations involves meticulous logistics from the DOF Office. In order to comply with evaluator anonymity, the faculty member must leave the classroom while the evaluation is taking place. After completion of the evaluation, a volunteer student hands over completed evaluations to the DOF Office, where results will be collected and completed evaluations distributed to the faculty.

## **2.3 Dissemination of Results**

In some institutions of higher education, DOF staff members review all evaluations to tabulate the results. Even if the institution has a Scantron system to process scannable forms [7], it takes a considerable amount of time to scan through all evaluations.

All evaluations are copied again after results are calculated so that the DOF Office may retain a copy and distribute the original copy to faculty members and other necessary parties. After faculty members submit their grades, they may obtain their evaluations from the DOF Office.

## **2.4 Use of Evaluation Results**

Institutions of higher education, such as Eberly College, use faculty evaluations as criteria for tenure and promotion [2]. Faculty members use evaluations individually to determine changes needed to improve courses and the classroom experience. Faculty members have numerous evaluations to analyze for every course. The time-consuming process sometimes breaks anonymity since it allows for faculty members to read handwritten responses. Even though faculty only receive evaluation results after grades are submitted, this could lead to a conflict of interest if a student attends another course with the same faculty during another semester. Some institutions have gone as far as typewriting all evaluation comments to avoid the possibility of faculty members identifying a student's handwriting!

Some faculty members leave campus by the time they are able to obtain the results of their course evaluations, which they will have to obtain during the following semester. Many faculty members plan their course materials during school breaks. Obtaining results for a previous semester is an inconvenience to faculty since they have already created course materials for the current semester based on the previous semester. If an evaluation highlights changes that should take place the next time the faculty teaches the course, it may be too late by the time the faculty receives the evaluation, which makes the whole process unproductive.

## **3. FINDING A SOLUTION**

The DOF Office took the first steps in contacting Institutional Technology (IT) for a solution. Addressing the challenge was an exciting opportunity for both parties since DOF Office staff members already had a solution in mind and were willing to see the idea to fruition with faculty members.

### **3.1 Requirements Gathering**

The DOF Office and IT met several times during hourly meeting sessions to discuss the project and plan. Over the course of a month, IT gathered enough requirements to begin development.

#### **3.1.1 Student Requirements**

According to the DOF Office, the CFA package should exist as a web-based system that contains all questions in the same order and divisions as the traditional printed evaluation form. Students should be able to anonymously submit course evaluations, pause an evaluation to continue at a later time without loss of data, and complete an evaluation when they answer all required questions. Once students complete an evaluation, they should be able to see which evaluations they have completed, but they should be "locked out" from evaluating any course for which they have completed an evaluation or after the evaluation deadline passes.

Even though the system provides a common login screen for all users, students should only be able to evaluate courses for which they have registered and should not be able to view evaluation results.

#### **3.1.2 Faculty Requirements**

Faculty members should be able to view current and archived evaluation results with all evaluation comments in a legible, printable format three weeks after classes end. Results should be tabulated with the following information included: total number of students in the course, total number of students who responded, total number of incomplete evaluations, and the date when the evaluation period ended.

Even though the system requires a common login screen for all users, faculty should only be able to view results for courses they teach and should not be able to evaluate courses; however, department heads should be able to view evaluation results for all faculty members who teach courses in their department.

#### **3.1.3 Administrative Requirements**

Staff members from the DOF Office are the administrators for the CFA package, so they should have the most access rights to the system. Even though DOF staff members should have the basic access rights of faculty members, they should also be able to view course evaluation forms as students see them and evaluation results for all faculty members. DOF staff members should be able to modify course evaluations before they are available for use and should be able to set beginning and end dates for course evaluation availability.

All users of the CFA package should authenticate with their existing account and the system should be able to recognize the user and the user's role upon login or should disallow login altogether.

Given the list of requirements, IT had to create a cost-effective, robust, scalable, flexible, adaptable, and secure solution.

## **3.2 The CFA Package is Born!**

After much research for solutions from other institutions that would fit the requirements set forth by the DOF Office, the best approach for our institution was to create our own CFA package, which we named CourseEval. A custom-made solution would not only fit the requirements for our institution, but could also be shared with other institutions facing similar needs.

#### **3.2.1 Strategic Planning**

Coordination is key among all parties involved to successfully develop a home-brewed system. Over the course of a month, the DOF Office and IT worked closely and reviewed all progress during development of the package. IT and the DOF Office held

weekly, hourly meetings to discuss each step and to review the timeline, if necessary. Due to close interaction, effective communication, and meticulous planning between the DOF Office and IT, the system was delivered for testing a week ahead of schedule.

### 3.2.2 The Framework

CourseEval uses three well-known open source software packages in its implementation: Apache Web Server 1.3.27 with secure module enabled [10], MySQL Database Server 3.23.42 [6], and Tomcat Application Server 3.3.1 [12]. CourseEval is implemented in the Java platform in a Mac OS X environment using the NetBeans Development Environment 3.6 [8]; however, it can run in any system that supports the required software packages previously listed.

### 3.2.3 Security

Given the requirements from the DOF Office, the security built into the package ensures that:

- Students cannot see each others' responses
- Students cannot view evaluation results
- Students cannot evaluate courses after the evaluation period
- Faculty cannot determine who responds to a question
- Faculty cannot view evaluation results until three weeks after classes end
- Passwords are protected during transmission
- Authentication is performed using Lightweight Directory Access Protocol (LDAP)

Most security requirements above are business processes built into the logic of the package itself; however, the last two requirements mandate careful planning and thorough testing so that users cannot enter the system without proper authorization. In our implementation, Apache SSL was used to proxy requests to and from Tomcat [13], effectively protecting usernames and passwords in network traffic.

LDAP authentication is built into CourseEval using the Java Naming Directory Interface (JNDI) to authenticate users. JNDI allows developers to create software packages that can easily communicate with directory services [5]. Property files specify which directory server and bindings to use as shown in Figure 1 [4]. CourseEval works with a role based access control, so each user who logs into the system will need to have an account in both the directory server and the CourseEval database.

```
# ldap.properties
ldap.host=ldapsystem.domain.edu
ldap.port=389
ldap.baseDN=cn=cn, dc=domain, dc=edu
ldap.givenName=givenName
ldap.sn=sn
ldap.uidAttribute=uid
ldap.managerDN=
ldap.managerPW=
```

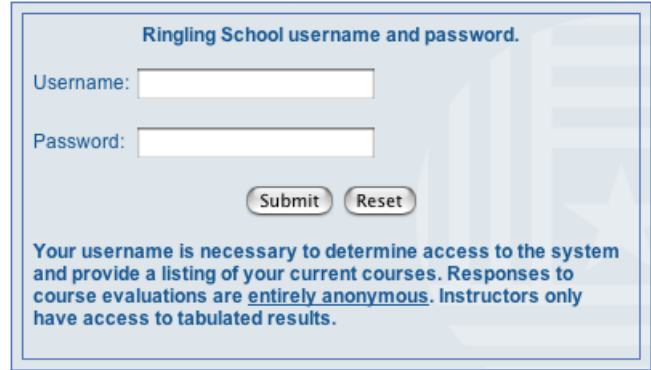


Figure 1. LDAP Configuration and Login Screen

Since both the web server and the database server reside in the same system, there is no need to take extra precautions beyond hardening the installation of the database server [6]. A property file specifies the connection to the database server and a user with proper access privileges as shown in Figure 2.

```
# rdbm.properties
jdbcDriver=com.mysql.jdbc.Driver
jdbcUrl=jdbc:mysql://localhost:3306/DB
jdbcUser=username
jdbcPassword=password
```

Figure 2. Database Server Configuration

### 3.2.4 Data Analysis

For the system to provide valuable results it has to tabulate the data contained in a database; however, does that mean it should tabulate data for similar courses, within a department, across departments, or individually for each course? For the sake of simplicity and arguments against comparing different courses with different teaching styles, materials and locations, CourseEval tabulates data for each individual course.

Each instructor has a separate view of the results for each course as shown in Figure 3. To maintain student anonymity, no student information appears in the system for faculty members, except for responses they have posted to evaluation questions. The DOF and the Vice President for Academic Affairs can view evaluation results for each faculty individually while department heads can view evaluation results for faculty members who teach courses for the department.

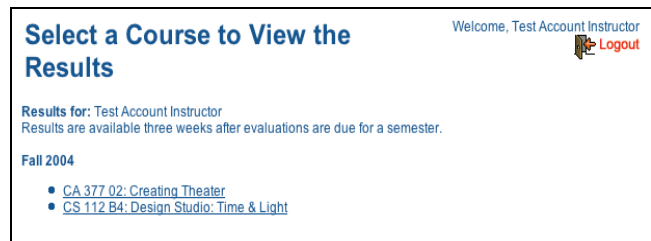


Figure 3. Faculty Results Selection Screen

Tabulated results consist of an average for each multiple-choice answer and a list of comments for each question that requires comments. The user can view all comments at once or by section as shown in Figure 4.

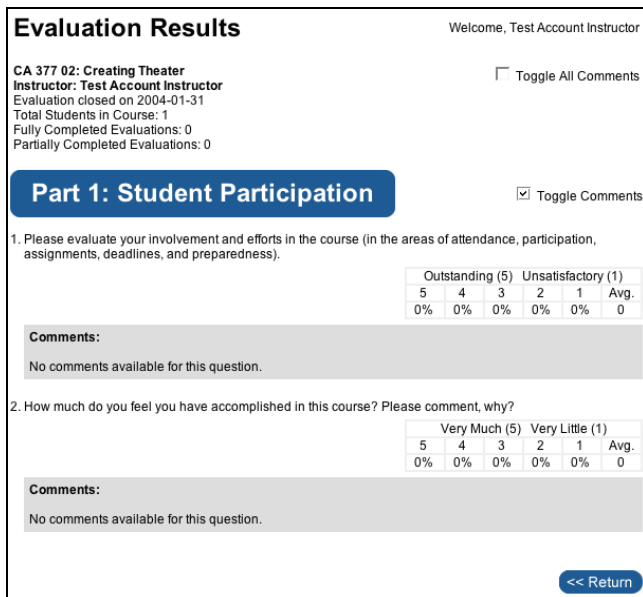


Figure 4. Evaluation Results Screen

### 3.2.5 Data Archives

CourseEval archives data per term. All questions, courses and responses for an evaluation are tied to the term, so even though questions may change for future assessments, previous assessment responses are still valid and do not conflict with newer question formats and responses. Upon login, appropriate users are able to choose previous terms to view older evaluation results.

### 3.2.6 Integration with Other Systems

The implementation of the CourseEval database includes tables from another software package, CourseWork. CourseWork is an open source course management system from Stanford University implemented on the Java platform [4]. Since CourseEval is based on the Java language and shares table schemas with CourseWork, it will be relatively simple to integrate CourseEval with CourseWork. CourseEval may also serve as a portal channel within a portal framework [9].

Currently, CourseEval is available through the DOF web site at <http://dof.ringling.edu>, a location well known to faculty members, and the Campus Portal at <http://my.rsad.edu>, which students and faculty members visit daily to access their e-mail.

### 3.2.7 Project Costs

Since development of CourseEval is internal to the institution, costs are minimal. An institution planning on using CourseEval will require the following:

- Administrative Staff Support (DOF, Academic Affairs)
- Web Developer
- Server-class system connected to the network
- Volunteer Testers

Administrative support is necessary to lead the project and guide IT in customizing the system for the institution's specific needs. A web developer knowledgeable with Java, HTML, SQL, and Java Script is essential to modify and extend the CourseEval package.

During development, our institution incurred the following costs:

- Administrative Support: 50 hours
- Web Developer: 165 hours
- Server-class system: already existing
- Volunteer Testers: negligible

Our production server is an existing Sun Fire 280R running the Solaris 8 operating system, so we were able to eliminate the cost of purchasing a new system. The web developer spent a month, full-time, working on the project until completion. Students and faculty volunteered for testing, which consisted of using the system instead of filling out an evaluation form for that term. Since the users performed the same task they would have performed with the traditional system, their cost is negligible. At our institution, the course evaluation process took about half an hour for students to complete during class time.

## 4. COMMUNITY INVOLVEMENT

Community inclusion during the early stages of development will foster successful acceptance of the finished product. Out of three groups: students, faculty and staff, we will discuss faculty members and students only since staff members are the catalysts for the project and will not use the system as much as students and faculty.

### 4.1 Faculty Inclusion

Faculty members may shy away from the idea of digital evaluations for numerous reasons. Faculty members will need to see a visual demonstration of the product and will require much reassurance from staff members involved in the project that the institution will treat the results in the same manner and with the same confidentiality as results obtained with the traditional method. A motivator for faculty is to be able to see the results from anywhere and to see concise results without the need to review numerous papers for each course.

Faculty members may have another concern involving continuity of responses. For example, when faculty members review the responses with the traditional method, they are able to view a set of responses for one individual. Individual responses allow faculty members to make conclusions if a specific response is very different from the rest of the responses. Since CourseEval tabulates all responses, the system eliminates the individuality of the responses. Since the system combines all responses for data tabulation, most faculty members who use the system still prefer it to the traditional method.

### 4.2 Student Inclusion

Students should be included early in the process as well. They will spread the word about the project and its goal. Student-workers may be the best volunteers for testing the system before its release. If student volunteers work in the divisions that implement the system, they are better able to articulate the project to others and foster support from the campus community.

While some students may be worried about the system maintaining anonymity, other students will prefer for their identity to be revealed to the instructors. A recommendation for students who prefer for their identity to be known is to place their name or initials inside each comment section of the evaluation.

## 5. TESTING A HOME-BREWED SYSTEM

If your system fails during its initial use, users will most likely perceive it as a failure and will probably not look forward to the next time they have to use it [1]. To avoid such occurrence, you will need to test your implementation under the same circumstances that will affect the final release.

### 5.1 A Call for Volunteers

Of course, it helps to have volunteers who support the system; however, you will want to take the difficult path and request for users who are not comfortable with the system to also volunteer for the test period. Your volunteers will require plenty of handholding since they will be working with a new, unfamiliar system. You will need to treat your new system like a newborn, and inform your volunteers, like concerned parents, of every situation, any change in the system and explain any issues that arise quickly and clearly. By the end of the test period, you will gain new alliances for the system and you will solidify your existing support base.

### 5.2 A Visit to the Classroom

A short step that you can take to “handhold” your volunteers is to visit them in the classroom while they are using the system. We intentionally asked for volunteers who meet in computer laboratories so that we could visit them and monitor the system at the same time. In our experience, few questions arose during these visits and volunteers raised no major issues. We received great feedback from the students, which we will share in the next section.

### 5.3 Feedback Collection

During the demonstration sessions for volunteer faculty and students, we gathered much helpful feedback that affected development of CourseEval. Since the demonstration sessions took place before CourseEval was tested, all recommendations were included in the version of CourseEval used during the test phase.

Students offered several important recommendations for inclusion in the campus-wide release of the system.

#### 5.3.1 *Spell Checking*

Several students would like to be able to spell check their course evaluation comments. Initially, the recommendation seems comical; however, faculty members could identify students through common spelling mistakes, which break anonymity in the system.

#### 5.3.2 *Full Evaluation Preview*

Several students recommend that a preview of the full evaluation will help them determine what to answer before they navigate through sections of the evaluation. Students will need to navigate back and forth between sections to make sure they have not answered similar questions with the same answer for an evaluation that contains multiple sections. A full preview will decrease the time it takes to fill out an evaluation with CourseEval.

#### 5.3.3 *Other Recommendations*

Other recommendations include the option to allow students to reveal their identity to faculty members, prevent users from losing responses if they have not saved a section and click on the “Back” button, and automatically e-mail users who have not completed their evaluations when the deadline approaches.

## 5.4 Feedback Response

Users should obtain immediate response to feedback. Withholding information from users will make them speculate about the system and could lead to unnecessary misconceptions about the system. As an anecdote, we received feedback from a student who was appalled that our institution would “succumb” to web-based course evaluations when the Internet is “inherently not secure.” After drafting a careful response with the DOF Office, we mailed a response to the user. Within minutes, we received a reply stating that the student was no longer opposed to the system and that she would have not opposed it in the first place had she known the facts. The student was under the misconception that anyone could easily obtain a list of users with their responses, that the system was not truly anonymous, and it caused the student to oppose the system. Clear communications will facilitate understanding and will foster support.

## 6. EASE OF ADOPTION

CourseEval uses three well-known open source software packages in its implementation: Apache Web Server 1.3.27 with secure module enabled, MySQL 3.23.42, and Tomcat 3.3.1. For development, the following packages are necessary: NetBeans 3.6, Sun Java SDK 1.3.1, and Apache Ant 1.6.1 [11]. CourseEval should work with later versions of the packages listed without modifications. A server-class system is required for deployment. CourseEval and more documentation are available at <http://www.rsad.edu/~lhermand/CourseEval>.

## 7. PLANNED FUTURE IMPROVEMENTS

We have already planned several major additions for the next version of CourseEval. Besides the recommendations from students and faculty members, the DOF Office requested additions for the Fall 2004 release of the project such as the “administrative view.” The administrative view, missing in the current release of CourseEval, will allow authorized Academic Affairs staff members to apply changes to course evaluations through the system, see a list of students who have not completed evaluations or a list of courses that are missing evaluations, and selectively open and close evaluations for specific courses. The addition will diminish the existing dependency on IT to make changes to course evaluations and alert users who have not completed evaluations.

All other recommendations from the students will be included in the future except the recommendation to allow students to identify themselves to faculty members. The counter-recommendation is for the student to mark all answers with their full name or initials. The DOF Office and IT want to prevent students from accidentally allowing the system to reveal their identity.

An initiative from IT is to look for ways to help staff and faculty members at the executive level to analyze the results. The DOF has to sift through close to three hundred faculty evaluations. Considering that each faculty member teaches at least two courses, this translates to six hundred evaluations, which is a daunting task even with CourseEval. Systems such as Criterion use heuristics to analyze essays. For information about Criterion, visit Educational Testing Service at <http://www.ets.org/criterion/index.html>. The technology would improve data analysis if it were able to create an executive summary for each faculty member or perhaps an entire department.

## 8. ACKNOWLEDGEMENTS

The authors would like to thank all students, faculty, and staff members that have made CourseEval a success at our institution and have helped by offering feedback as to what helps them work more efficiently. Without their support and patience, CourseEval would be nonexistent. Thanks to our colleagues for their support and valuable input. We would also like to thank Apple Computer for their generosity, adherence to standards, and valuable documentation; and the open source projects from the Apache Software Foundation and MySQL AB, which allow educational institutions to complete cost-effective works such as the one presented in this paper. Thanks to Sean Conkin, Kris Pegah, and Tom Wetherby for their constant support and understanding.

## 9. REFERENCES

- [1] Earnshaw, A. C. (2004). *Why Do Large IT Projects Fail?*, 2004, from <http://www.infoms.com/wp-large.htm>
- [2] Eberly College of Arts and Sciences. *Guidelines for Annual Faculty Evaluation, Merit Determination, Promotion and Tenure*, 1998, [http://www.as.wvu.edu/forms/guidelines\\_for\\_annual\\_faculty\\_ev.htm](http://www.as.wvu.edu/forms/guidelines_for_annual_faculty_ev.htm)
- [3] Felder, R. M. *What do they know, anyway? II. Making Evaluations Effective*, 1993, <http://www.ncsu.edu/felder-public/Columns/Eval2.html>.
- [4] Horstmann, C. S. and G. Cornell. *Core Java 2: Volume 1- Fundamentals*. Prentice Hall, Palo Alto, CA, 2003.
- [5] Lee, R. and S. Sligman. *JNDI API Tutorial and Reference: Building Directory-Enabled Java Applications*. Addison Wesley, Palo Alto, CA, 2000
- [6] MySQL AB. *Installing MySQL*, 2004, <http://dev.mysql.com/doc/mysql/en/Installing.html>
- [7] Scantron. *Scantron Scanner – Overview*, 2004, <http://www.scantron.com/products/scanners/index.asp>
- [8] Sun Microsystems. *General NetBeans Frequently Asked Questions*, 2004, <http://www.netbeans.org/kb/faqs/general.html>
- [9] Sun Microsystems. *JSR-000168 Portlet Specification (Final Release)*, 2000, <http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html>
- [10] The Apache Software Foundation. *Compiling and Installing Apache 1.3, Apache Documentation Project*, <http://httpd.apache.org/docs/install.html#apaci>, 2003.
- [11] The Apache Software Foundation. *Installing Ant*, 2004, <http://ant.apache.org/manual/installlist.html>
- [12] The Apache Software Foundation. *Using Tomcat 3.3.*, 2001, [http://jakarta.apache.org/tomcat/tomcat-3.3-doc/tomcat-ug.html#using\\_tomcat](http://jakarta.apache.org/tomcat/tomcat-3.3-doc/tomcat-ug.html#using_tomcat)
- [13] The Jakarta Project. *Proxy Support How-To*, 2002, <http://jakarta.apache.org/tomcat/tomcat-4.0-doc/proxy-howto.html>.
- [14] Yun, G. W. and C. W. Trumbo. *Comparative Response to a Survey Executed by Post, E-mail, & Web Form*, 2000, <http://www.ascusc.org/jcmc/vol6/issue1/yun.html>