In this issue:

The Role of the Web Server in a Capstone Web Application Course

Karthikeyan Umapathy
University of North Florida
Jacksonville, FL 32224 USA

F. Layne Wallace
University of North Florida
Jacksonville, FL 32224 USA

Abstract: Web applications have become commonplace in the Information Systems curriculum. Much of the discussion about Web development for capstone courses has centered on the scripting tools. Very little has been discussed about different ways to incorporate the Web server into Web application development courses. In this paper, three different ways of incorporating the Web server are discussed: shared Web server (minimal student control), managed Web server (configuration control), and controlled Web server (full student control). This paper argues that capstone courses oriented towards Web applications development should provide certain amount of Web server control to students as it is an important component of any Web application.

Keywords: Web server, server configuration, Web applications, capstone courses

This paper was in the 2009 cohort from which the top 45% were accepted for journal publication. Acceptance is competitive based on at least three double-blind peer reviews plus additional single-blind reviews by the review board and editors to assess final manuscript quality including the importance of what was said and the clarity of presentation.

The Role of the Web Server in a Capstone Web Application Course

Karthikeyan Umapathy
k.umapathy@unf.edu

F. Layne Wallace
lwallace@unf.edu

School of Computing
University of North Florida
Jacksonville, Florida 32224 USA

Abstract

Web applications have become commonplace in the Information Systems curriculum. Much of the discussion about Web development for capstone courses has centered on the scripting tools. Very little has been discussed about different ways to incorporate the Web server into Web application development courses. In this paper, three different ways of incorporating the Web server are discussed: shared Web server (minimal student control), managed Web server (configuration control), and controlled Web server (full student control). This paper argues that capstone courses oriented towards Web applications development should provide certain amount of Web server control to students as it is an important component of any Web application.

Keywords: Web server, server configuration, Web applications, capstone courses

1. INTRODUCTION

The high potential rewards of conducting business activities over the Web have enticed many organizations to adopt e-Business technologies. The critical imperative of maintaining a competitive and sustainable market advantage has forced organizations to invest in e-Business infrastructures. Essential e-Business infrastructures include technologies for the development of Web applications (Papazoglou & Ribbers, 2006). These Web applications allow the exchange of large volumes of information with customers and partners globally.

As a consequence of the growth of e-Business, Web technologies have become an established part of the Information Systems (IS) curriculum (Yue & Ding, 2004). Even the traditional analysis, design, and implementation capstone courses have shifted to Web applications (Yan & Fang, 2005). Most of these capstone courses focus on developing Web applications using scripting languages (Yue & Ding, 2004) and database driven Web applications (Janicki, Fischetti, & Burns, 2005). However, an important aspect of Web application development is not addressed in many of these capstone courses: installing, configuring, and optimizing Web servers. In this paper, we present three different ways the Web server can be incorporated into a capstone course oriented towards Web applications.

2. COMPONENTS OF WEB APPLICATIONS

Web applications, in a boarder context, can be defined as any software application that runs on the Internet where a person uses a Web browser on their computer to run the application residing on the Web server (Chen & Heath, 2001). The general architecture of Web applications is usually a three-tier architecture, see figure 1. Correspondingly,

key components of Web applications are Web clients, Web servers, and gateways (Papazoglou & Ribbers, 2006). Gateways are external resources such as legacy applications and databases that are typically leveraged by Web applications.
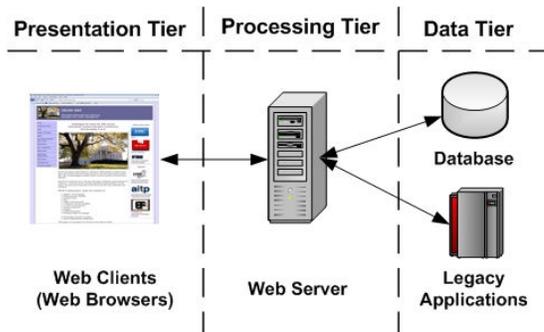
Figure 1: Three-tier Web architecture

Web clients (usually Web browsers) provides the user interface for accessing the application with a primary function to accept and validate user input and present results received from a Web server (Hadjerrouit, 2001). Through Web clients, users can communicate with Web servers using Internet protocols to access business data and logic (Papazoglou & Ribbers, 2006).

The Web server manages and distributes files such as Web pages and other multimedia files to be published on the Web (Papazoglou & Ribbers, 2006). The main function of a Web server is to process user requests by coordinating access to business logic and returning Web pages containing static and dynamic information from existing applications and databases (Hadjerrouit, 2001). The Web server provides server-side programming capability to develop the business logic part of the application, to process client requests and to produce a response Web page that is returned to the client (Chen & Heath, 2001). The Web server also provides caching, remote administration, directory, and security services (Papazoglou & Ribbers, 2006).

3. WEB APPLICATIONS AND CAPSTONE COURSES

The success of e-Business applications depends upon successful Web application development and successful integration of the Web server through which applications are made accessible (Papazoglou & Ribbers,

2006). Key aspects for the successful development of Web applications are understanding the needs, requirements, and operating environment of the system; analyzing and designing the application using appropriate Web architectures; and constructing the application using appropriate Web programming languages (Ginige, 2002). The above identified important issues, as per IS curriculum guidelines ((Gorgone et al., 2002), pg:13-14) and Information Technology (IT) curriculum guidelines ((Lunt et al., 2008), pg:17-18), are considered to be important topics for IS and IT undergraduate students.

A review of the literature on capstone courses that are oriented towards Web applications indicates that the above identified issues are faithfully addressed. For example: understanding user needs and systems requirements (Janicki et al., 2005), designing Web systems using appropriate architectures (Yap & Loebbecke, 2005) and database-driven Web applications using Java (Wang, 2003) and using .Net (Frydenberg, 2008; Wallace & Wolf, 2008). There have been studies on factors considered for selecting server side programming languages for teaching capstone courses (Sandvig, 2007), assessing student's performance for Web application courses taught through online method (Sissom, Shih, & Goro, 2006), and integrating essential concepts of data-driven Web applications into a capstone course (Kovacs & Baugh, 2009).

Key aspects of successful Web server management are installing, configuring, and maintaining the Web server (Winer & Maniotes, 2001). Association for Computing Machinery (ACM) computing curriculum recommendations suggests that Web server management is an important skill set for IT students than for IS students ((Shackelford et al., 2005), pg:28). Moreover, both IT and IS curriculum guidelines have larger focuses on the developmental aspects of Web applications in comparison to Web server management.

With increasing use of e-Business technologies by organizations, skills required to develop Web applications and to take full functional advantage of Web servers are important. However, regarding the above ACM curriculum guidelines, a recent search of computing education journals (such as Information Systems Education Journal

(ISEDJ), Journal of Information Systems Education (JISE), and Journal of Computing Sciences in Colleges) showed that most Web applications courses have an emphasis on the developmental aspects of Web application (including database-driven applications) and either completely ignore or weakly address aspects of Web server integration. In era of Web-based and other advanced Internet technologies, Web server integration should also be a crucial part of any IS and IT curriculum.

4. INCORPORATING A WEB SERVER INTO CAPSTONE COURSE

The Web server can be incorporated into a capstone course in one of three ways based on the amount of Web server control given to the students. Three of these approaches could be: 1) a shared server (minimal student control), 2) a managed server (configuration control), or 3) a developer controlled server so that the students control how the server is built and which components are embedded.

Shared Web Server

The shared server is the scenario in which a single Web server is shared by the entire class. This means that each student or group must use a Web server with the same configuration (regardless of their individual projects), the same access, the same embedded components, and common security issues. Each group of students must access the Web server through the same URL and separate themselves by means of separate Web content subdirectories. If the students are using scripting languages embedded in the Web server then all students or teams must use the same versions of the tools with the same basic configuration with no means of incorporating or optimizing the Web server with their project. There are, however, some advantages to this approach. No additional class time is necessary to discuss the Web server internals or configuration, simpler Web administration, more time spent in the class talking about the immediate development process, and simpler student or group accounts.

Managed Web Server

The second way a Web server can be incorporated into a course is by deploying a Web server for the students where the students or teams are provided with separate servers on the same computer. These servers may be built for the students with different characteristics depending on the needs of the students or teams including different scripting tools either separate from or embedded in the Web server. The students or teams would each have a separate network port for access and, through the Web server runtime configuration file, separate access characteristics for their projects. Additional class time would be necessary to discuss the Web server configuration file. The primary advantages of this technique are more control and flexibility over a specific Web server for each student or group, individualized usage logs, and a greater understanding of how Web servers process user requests and business and data logic.

Developer-controlled Web Server

The third way to incorporate a Web server into a course would be to have the students build and/or install the Web server themselves. The students or groups could decide which components they wanted in the Web server, which tools to use to build the server, and, potentially, which specific Web server to use. Naturally, class time would be needed to explain the installation and configuration of the Web server. The primary advantages of this technique are a fuller understanding of the computer system as a whole instead of individual software processes, an example of a large system along with the installation and configuration components, and complete control over the Web server selection and services it provides.

5. PRACTICAL EXPERIENCES

All three approaches were evaluated using different classes over three years. The students were enrolled in a two-semester application capstone development course which required team participation. The course started with analysis and initial design in the first semester and final design and implementation in the second semester. The Web server was Apache running on a Sun Microsystems Enterprise E450 multiuser computer with the Solaris operating system and was used for all instances of the course. The machine was dedicated to the course and had no other users on it. Each team member was

required to have an active development role during both semesters of the course and a responsibility log was maintained by the team leader. The teams met with the instructor in person and electronically during each semester to make sure that progress was being made.

Shared Web Server

For the first approach students were required to use a shared Web server. The instructor installed and configured the Web server. All teams were given an account on the class computer and separate directories were established for each team in the Web space. Each team was given ownership of their Web space. Considerable class time was devoted to discussions about securing each team's project space through directives for each subdirectory and reading the log files which contained data from all of the teams' projects. Students were also given an account on the class' Database Management System (DBMS).

In the debriefing at the end of the semester, the students commented that the lack of control over the Web server made the development effort more difficult than they felt was necessary. One example of this was apparent in that the permissions of the Web content directories were set for each team, however, the owner of the Web server process had to have permission to access these directories.

Managed Web Server

The next course offering used individual Web servers. Each team had an account on a computer dedicated to the course. The instructor built and configured the Web server in each team's home directory. PHP was the scripting language and was embedded in each Web server. After the teams were formed, one class period was spent discussing the Web server configuration file and the implications for each entry. Students were shown how to secure the entire Web space and individual subdirectories within their Web space. The log files for each team were specific to their individual Web servers so that much of the functionality of their application audit modules were performed by the Web server.

In the debriefing at the end of the semester, the students were very pleased with the control they had over the Web server and felt that the control helped them develop the application more completely than the typical class projects assigned for other courses. They could quickly read their logs for fast debugging and change the server's configuration for different access and reporting levels.

Developer-controlled Web Server

The last course offering used the technique where the teams each installed and configured their own Web servers. The teams were allowed to choose the Web server and scripting language they felt best fit the needs of the project. This was intended to give the students a more realistic experience where they could make the choice of the three-tier Web architecture components. One class period was dedicated for discussion about general principles of Web servers and the implications of different configurations. All but one team decided to use the Apache server with embedded PHP while the remaining team used the Apache Tomcat server with Java. Many office hours were spent discussing various compiling issues for the server and the scripting languages. Some groups tried downloading a precompiled Apache Web server for Solaris but found that they lost a great deal of control about where the server components were stored, which components were included in the server, and how the server could be configured.

In the discussions at the end of year, the students seemed to be of mixed minds as to the advantage of installing their own Web servers. Most felt that they spent a large amount of time installing a tool instead of developing a software application project. Yet, the majority of the students also mentioned that the additional control of the production environment made their software projects more complete. The additional security and audit controls were mentioned by all teams. Another common comment was that they learned a great deal about computing concepts that they hadn't gotten in other courses. While each class was required to package their final project for distribution and installation, the teams who installed and configured their own Web servers had fewer packaging and installation problems. Table 1 (see appendix A) provides a summary of the

advantages and disadvantages for the three approaches described in this paper.

6. CONCLUSIONS

Based on student comments and project quality, incorporating a Web server's configuration into a software project development course is beneficial for students in several ways. The students appreciated the additional control over the Web server's function, the server's logs, and the software system access while not having to spend time installing the Web server. Naturally, a production environment outside the educational setting may constrain developers to a specific Web server configuration. However, students in an educational setting seem to gain additional understanding about the software development process from having a certain amount of control over the software environment (Lowe, 2003). The students who installed and configured their own Web servers also incorporated some of the features of the Web server into their projects (for example, the configuration file structure).

Multiple Web servers do present an increase in system administration and security activities for the faculty. For each new security or functional update to the Web server, each software server will have to be updated and tested. Additionally, examining the logs for system security breaches becomes more difficult since the logs are not maintained in a central location.

Currently, most capstone courses oriented towards Web application development have a larger emphasis on the development of languages. The Web server is an important aspect of Web application and should receive a certain amount emphasis in capstone courses. We conclude, with the argument that in order to provide comprehensive experience for students, capstone courses should include Web server integration as a part of student projects. Such a measure would be helpful in strengthening program learning outcome of providing comprehensive knowledge and skills on e-Business technologies.

7. REFERENCES

Chen, J. Q., & Heath, R. D. (2001). Building Web Applications: Challenges, Architectures, and Methods. *Information Systems Management, 18*(1), 1 - 12.

Frydenberg, M. (2008). Slickr: A Multi-Tiered Web Development Capstone Project Using Databases, Web Services, and AJAX. *Information Systems Education Journal, 6*(37), 1-15.

Ginige, A. (2002). *Web engineering: managing the complexity of web systems development.* Paper presented at the International Conference on Software Engineering and Knowledge Engineering, Ischia, Italy.

Gorgone, J. T., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L., & Longenecker, H. E. (2002). IS 2002 - Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Retrieved July 7, 2009, from http://www.acm.org/education/education/curric_vols/is2002.pdf

Hadjerrouit, S. (2001). Web-based application development: a software engineering approach. *ACM SIGCSE Bulletin, 33*(2), 31-34.

Janicki, T. N., Fischetti, D., & Burns, A. T. (2005). Incorporating Real World Projects and Emerging Technologies into One MIS Capstone Course. *Information Systems Education Journal, 5*(24), 1-8.

Kovacs, P. J., & Baugh, J. M. (2009). Merging Object-Oriented Programming, Database Design, Requirements Analysis, and Web Technologies in an Active Learning Environment. *Information Systems Education Journal, 7*(52), 1-8.

Lowe, D. (2003). Web system requirements: an overview. *Requirements Engineering, 8*(2), 102-113.

Lunt, B. M., Ekstrom, J. J., Gorka, S., Hislop, G., Kamali, R., Lawson, E., et al. (2008, November). Information Technology 2008 Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. Retrieved July 7, 2009, from http://www.acm.org//education/curricula/IT2008%20Curriculum.pdf

Papazoglou, M. P., & Ribbers, P. (2006). *e-Business: Organizational and Technical Foundations* West Sussex, England: John Wiley & Sons, Ltd.

Sandvig, J. C. (2007). Selection of Server-Side Technologies for an E-Business Curriculum. *Journal of Information Systems Education, 18*(2), 215.

Shackelford, R., Cross, J. H., Davies, G., Impagliazzo, J., Kamali, R., LeBlanc, R., et al. (2005, 30 September). Computing Curricula 2005: The Overview Report. Retrieved July 12, 2009, from http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf

Sissom, J., Shih, S., & Goro, T. (2006). A Low-Cost Remote Lab for Internet Services Distance Education. *Journal of Systemics, Cybernetics and Informatics, 4*(4), 1-4.

Wallace, D. C., & Wolf, J. R. (2008). Teaching ASP.NET in an Information Systems Curriculum. *Information Systems Education Journal, 6*(41), 1-12.

Wang, M. (2003). E-Business Application Development With Java Technology And Oracle: The Fortune Invest Inc. Case.

Journal of Information Systems Education, 14(3), 293-300.

Winer, C. R., & Maniotes, J. (2001). *The Two- and Four-Year Internet/Web Programs at Purdue University Calumet.* Paper presented at the Information Systems Education Conference, Cincinnati, USA.

Yan, Y., & Fang, X. (2005). Technology Adoption in E-Commerce Education: A Case Study. *Information Systems Education Journal, 3*(40), 1-8.

Yap, A. Y., & Loebbecke, C. (2005). A System for Teaching MIS and MBA Students to Deploy a Scalable Database-driven Web Architecture for B2C E-Commerce. *Information Systems Education Journal, 3*(7), 1-20.

Yue, K.-B., & Ding, W. (2004). Design and evolution of an undergraduate course on web application development. *ACM SIGCSE Bulletin, 36*(3), 22-26.

APPENDIX A

Table 1. Advantages and disadvantages of different ways of incorporating a Web server into a capstone software development course

Advantages	Disadvantage
Shared Web Server – single Web server shared by the entire class	
• Less class time will be spent on discussing Web server configuration and more time spent on the application development.	• The same configuration will be used by all groups with no option to optimize it. • A lack of understanding of the Web server processes is a missed educational opportunity. • A lack of control can create difficulties with application development particularly for security and performance issues. • Installation and packaging problems for distributing the application are common.
Managed Web Server – different Web servers configured to satisfy needs of each student groups	
• Students will have more control and flexibility over the configuration of the server. • Students will get greater understanding of Web server processes. • Access to log files can be helpful for debugging and creating better configuration.	• Adequate class time is required to discuss how to Web server configuration file. • Installation and packaging problems for distributing application can occur.
Developer-controlled Web Server – students with complete control of selecting, deploying and configuring the Web server	
• Complete control of Web server provides greater understanding of Web server management issues. • Fewer installation and packaging problems arise for distributing the application.	• Adequate class time required to explain how to select server, install, and configure Web server. • More time spent on installing and configuring the server means a loss of time for application development.