

Implementing a Dynamic Database-Driven Course Using LAMP

Joseph Packy Laverty
laverty@rmu.edu

David Wood
wood@rmu.edu

John Turchek
turchek@rmu.edu

Computer and Information Systems,
Robert Morris University
Pittsburgh, PA 15219, USA

Abstract

This paper documents the formulation of a database driven open source architecture web development course. The design of a web-based curriculum faces many challenges: a) relative emphasis of client and server-side technologies, b) choice of a server-side language, and c) the cost and efficient delivery of a dynamic web development, database-driven platform. This paper reviews alternative dynamic web development, database-driven platforms and presents a case study of integrating LAMP, an open source dynamic web data-base driven solution, in an Information Systems Curriculum. Three sections were presented over a three-year period. Information concerning course content, instructional delivery methods, alternative LAMP technological infrastructures, student retention and performance are also discussed.

Keywords: LAMP, WAMP, PHP, Apache, MySQL, Dynamic web pages, Open source, Web development, Database-driven web sites

1. INTRODUCTION

In response to the high demand from industry, teaching web development and programming has become an important component of the CS/IS curriculum (Wang, 2009.) Janicki, Gkown, Kline, & Konopaske (2004) conducted an exploratory survey which provided evidence that employers are increasingly interested in both proprietary and open-source dynamic, database-driven web development skills. Employers indicated that skills with Windows or Linux, VB.NET/ASP.NET and SQL Server were desirous. The study further indicated programming skills for entry-level hires remains at a high level as compared to their previous surveys.

The design of IS web-based curriculum faces several challenges. These design challenges include: a) relative emphasis of client and server-side technologies, b) choice of a server-side language, and c) the cost and efficient delivery of a dynamic web development, database-driven platform. This paper reviews alternative dynamic web development, database-driven platforms and presents a case study of integrating LAMP in an Information Systems Curriculum. The results of alternative LAMP technological infrastructures, instructional delivery methods, and student retention and performance were analyzed and discussed.

2. BACKGROUND AND RELEVANT LITERATURE

Web Based Curriculum

Perry (2002) listed several components necessary to support a dynamic, database-driven (DDD) web site. A dynamic website has five major platform components: the operating system, the web server, the application server, the database and the programming/script language. Markup or programming languages represent only one component that is required to support a dynamic, database-driven web-site.

Chung & McClane (2002) listed a diverse selection of languages that serve as a basis for web-based curricula. These languages include: markup languages ((X)HTML and XML), style sheet languages (CSS, XSL), client-side languages (JavaScript), server-side embedded languages (PHP, JSP, ASP, CFML), and server-side high-level languages (Java, ASP.NET). This list illustrates the number of infrastructure tiers and alternative languages which increases the complexity of designing a web-based curriculum.

Chung & McClane's (2002) case study was based on a course that included various client-side technologies and server-side Java and JavaBeans applications. Their conclusion was that their approach was successful because: a) many students had a first-level program language course in Java, b) the low cost of open source software, c) and students were able to install and administer their enterprise environment. On the other hand, the Chung & McClane study (2002) also reported challenges for student's access to lab computers, lack of documentation, and extra work for the instructors.

A variety of dynamic, database-driven platform solutions exist to support E-Commerce and Content Management System (CMS) web sites. Frequent comparisons are made between two popular dynamic web platforms: LAMP (Linux, Apache, MySQL, and PHP) and WISA (Windows, IIS, SQL Server, and ASP.NET) (Perry, 2002, Web Master Tips, 2006.) Other solutions have included other dynamic program/script languages, e.g., Java, Java Server Pages (JSP), and Cold Fusion Markup Language (CFML), Database Management Systems, e.g., PostgreSQL, Oracle, DB2, and Application Servers, e.g. Tomcat, JBOSS, WebLogic, and WebSphere.

Within the structure of a DDD platform there is also a need to distinguish between HTTP web servers and application servers. Web servers support delivery of static web page content using the HTTP protocol. Examples of HTTP web servers include Apache and IIS. On the other hand, application servers will provide an environment that will execute server-side applications and provide database connectivity to a data base management system. Apache and IIS can load additional modules that provided the application server function. Other application servers are independent of the web server, e.g., JBOSS, Tomcat, etc. Application servers and dynamic server-side programming languages are at the core of dynamic, database-driven web sites.

While there may exist many newer and easier-to-use dynamic web platforms, e.g., Ruby on Rails, ASP.NET and J2EE, the popularity of LAMP continues to grow (Learn Computer, 2010.) Builtwith.com (2010) reported that PHP was in active use by more than 2.9 million web sites and 33% of the top one million active web sites. Apache was the most popular web server representing 55.8% of all public web servers. In 2008 then over 11,700 registered PHP projects listed on SourceForge.net and other high-profile applications like Face Book and Wikipedia (Cholakov, 2008).

Using Lamp as a Dynamic, Database Driven Platform

There are many reasons why open source software is popular. According to the Open Source Initiative (2010), "Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in." Dionisio, Dickson, August, Dorin, & Toal (2007) proposed that the characteristics of the Open Source Culture should be reflected in the teaching framework presented in all four years of an undergraduate, computer science curriculum. Several sources have listed the advantages of using LAMP and open source software as follows:

- Open Source Licensing (no cost) or Large Scale Commercial License Alternatives (Scalability)
- Non-proprietary. User communities set development goals and provide free support. Faster feature development

- Popularity and Wide Deployment
- PHP is easier to learn than Java, Java Server Pages, and ASP.NET
- PHP can be coded in either a procedural or object-oriented style.
- PHP can be used on a variety of operating system platforms and web server (cross-platform compatibility)
- PHP is faster than other scripting languages, EJB, Java Servlets and comparable to ASP.NET. Differences in Java performance decreases in three-tier environment.
- PHP, Linux, Apache and MySQL tend to be very stable and do not change radically between versions.
- PHP supports a wide variety of standard and object-oriented databases.
- MySQL supports stored procedures and triggers
- Since PHP may be hard-coded rather than generated by a WYSIWYG editor, better code may be created.

(Converse & Park, 2004, Web Master Tips, 2006, Learn Computer, 2010. My Solutions, 2009, Misfit Greek, 2010, Sheldon & Moes, 2005)

Open source versions of LAMP may provide a practical way for students to experience open software within the curriculum. However, one must be careful to distinguish that LAMP is both an open source platform and also widely available as proprietary platform from Red Hat, IBM and other vendors. Several sources have presented some of the disadvantages of PHP as a programming language:

- PHP variables are loosely typed, which can lead to some problems that are difficult to detect.
- Inconsistent case rules: PHP variable names are case-sensitive while function name are not.
- Global variables may be changed by hackers in the HTTP header.
- PHP does not require modular or object-oriented programming, which can lead to poor programming techniques.

- While PHP code may be compiled, there is no support for multithreaded operations or asynchronous execution
- Exception handling was only introduced in later versions and is not backward compatible
- Scalable options, e.g., clustering, replication of distributed databases, partitioning, failover, etc. are only available at a cost using proprietary versions, e.g., Red Hat Enterprise Linux, MySQL AB

(Web Master Tips, 2006, Cholakov, 2008, MySQL Enterprise Server 5.1, 2010, Cecchet, Chand, Einikety, Marguerite, & Zwaenepoel, 2010, Learn Computer, 2010, My Solutions, 2009, Sheldon & Moes, 2005, Misfit Greek, 2010.)

Some of these disadvantages reflect some limitations found in many open source software components. Proprietary LAMP alternatives provide better and easier-to-use administration tools, scalability options, fault tolerance, technical support, and development tools. From an IS curriculum point-of-view, the open source versions of LAMP are more than adequate. Many practitioners also use open-source alternatives for many limited scale, internal projects.

Approaches of Using Lamp in the IS Curriculum

LAMP may be used demonstrate dynamic web sites to a variety of audiences. Harris's PHP and MySQL book (2004) uses games like poker and dice to present basic programming structures, e.g., sequence, selection and iteration, and a simple database. Lecky-Thompson (2008) adds slightly more depth for beginner programmers to develop a simple content management.

LAMP also provides an excellent platform to provide a capstone course. LAMP text books provide many projects like: Online Address Books, Discussion Forums, Online Storefront and Shopping Carts (Meloni, 2008). PHP may also be used to present more advanced web applications, e.g., Ajax (Ballard & Moncur, 2009), application security (Shifflet, 2005), and web services (PHP.net, 2010). Finally, Lecky-Thompson (2005) uses PHP to cover object-oriented project management, analysis, design, application development, testing, and deployment.

1. Case History of LAMP-Based Course in a IS Curriculum

The Need for a Dynamic Database-Driven Web Course

In 2007, the Computer Information Systems Department conducted a review its web development curriculum. It was discovered that all existing courses related to web development were based on the Windows platform. Courses offered included: Web Page Development (XHTML, HTML, CSS and JavaScript), VB.NET and ASP.NET, and Windows Server Administration. The only course based on an open source alternative was Linux System Administration. Further analysis of the curriculum and course content indicated that there were several deficiencies in the coverage of many components of a dynamic, database-driven platform. Some deficiencies cited included: web server and application server administration, database server administration, and inadequate coverage of enterprise and dynamic web applications.

Many IS curriculums are faced with the challenges of balancing currency with the content of existing courses with a new required or elective course. While the demand of industry may indicate a need for an Open Source Dynamic Database-Driven course, whether this proposed course be a developed as new course or should an existing course content be revised? For example, some faculty members questioned the emphasis on traditional application interfaces and algorithms, e.g., command line and data structures, at the expense of popular enterprise (multi-tiered) dynamic web applications. The debate continues.

In the fall of 2007, the CIS department decided to introduce a new elective course titled "Open Source eCommerce Development (LAMP)." The original objectives of this course were to: a) increase student awareness of open source technologies, b) present dynamic web, database-driven application development from a multi-tiered and administrative perspective, c) minimize course prerequisites, and d) be capable of delivery in a 15-week online instructional format.

This course's outcomes and topic coverage were designed to mirror those of existing courses, Linux System Administration, Database Management System, Introduction to Web Development, and Advanced Web Development (See the Appendix Table 1 for a comparison.)

Blackboard was used to provide online content and testing. A wide variety of detailed, instructor-developed tutorials were presented. Students were required to complete a semester application program and administration project. Experience using (X) HTML tables and forms was recommended. Course PHP application assignments did not require object-oriented or intermediate-level programming experience. Local and remote LAMP/WAMP alternatives were provided. Supplementary on-ground help sessions were provided.

Except for PHP, other course content areas were designed to be presented at an introductory-level. Database table layouts were provided to students. Each student was required to code the necessary MySQL code to implement the table layouts design and then insert test data.

Emphasis was placed on PHP as being the dynamic application interface between the web server, dynamic HTML content and the database. Several PHP code templates were provided to students. No GUI PHP editors or code generators, e.g., Eclipse or Zend Studio, were used. MySQL GUI administrative tools, e.g., PHPMYAdmin, were not used to create the database schema or enter test data.

Local LAMP/WAMP

Early in the semester students were required to install WAMP or LAMP on their personal computer or server. All students had chosen a Windows-based solution, e.g., WAMP, for their personal computers. Advantages of local LAMP/WAMP installation included: a) experience in installing and debugging LAMP or WAMP installation, b) students had access to all PHP, MySQL and Apache configuration files (httpd.conf, my.cfg and php.ini), and c) students could use Windows-based editors and other utilities for which they may be more comfortable using. Disadvantages of local LAMP/WAMP installation included: a) the Linux Operating System was not in use, b) remote server access and administration may not be emphasized, c) limitations for instructor verification, and d) limitations for student collaboration.

Course Public LAMP Web Site

Considering the advantages and disadvantages of local installation, a remote student-shared LAMP server was provided. Students were encouraged to use PuTTY and WinSCP open

source utilities for SSH terminal access and file transferring. No Telnet or FTP was provided. Students were not required to install or configure any LAMP resource on the student-shared server, e.g., Linux permissions, httpd.conf, my.cfg and php.ini. Each student was provided a Linux user account and home directory. Each student was provided an individual Apache name-based virtual web server and a MySQL database with appropriate administrative permissions.

Using an Apache name-based virtual web server each student was assigned a separate document root directory. Each student web developer was assigned the appropriate Linux file permissions by the instructor (root). Appropriate Linux file permissions to permit web content to be viewed by the public or other students were also assigned. No extra Apache authentication, authorization or access control restrictions were used in the initial setup. PHP script or SQL code/logic is not displayed or accessible in the internet browser window. As compared to static (X) HTML web pages, the intrinsic nature of dynamic, database-driven web development improves student collaboration while protecting the academic integrity of the source code.

Students could test or view their own or other student's web sites by entering an individual domain name into their internet browser address bar. An advantage of using Apache name-based virtual servers is that only one public IP address is required. The university's IT Services department entered each name-based subdomain name into the university's DNS server.

Results of Course Offerings

Two different sections of a 15-week online-format of Open Source eCommerce Web Development (LAMP) were offered and completed in Fall/2007 (n=15) and Fall/2008 (n=10). On-ground instructor-led, voluntary student-help sessions were scheduled on a weekly basis. Student retention and performance for both sections were disappointing. A detailed review of student tests, assignments, instructional materials, text, and student background was conducted. It was determined that the only significant factor for successful student retention and performance was their voluntary attendance at the weekly on-ground student-help sessions. It was decided that several changes were necessary.

A third section of Open Source eCommerce Web Development (LAMP) was offered and completed in Fall/2009 (n=11). The instructional delivery method was changed to a hybrid approach using a nine-week format. An on-ground class meeting was divided into two parts: an instructor demonstration/lecture and a student-help session. The course content, Blackboard support materials, tests, and assignments remained substantially unchanged. There was a change in the required textbook. At the instructional-level, only the delivery method and the textbook were significantly changed. At the technological infrastructure level, there was a change from the Apache name-based web server approach to a VMWare-virtualized server. However, these changes in the technological infrastructure were transparent to the individual student.

Student retention and average assignment performance of students for the Fall/2009 section improved by 37% and 61% respectively. The small sample sizes for each course section limited statistical analysis for level of significance.

In addition, a section of Linux System Administration was also redeveloped in a 9-week hybrid format, and scheduled subsequent to the new course. Putting both of the courses together in the same term seemed to increase student interest and enrollment.

Virtualization of the LAMP Infrastructure

The technological infrastructure that supported the student's remote access to LAMP changed with the third section offered in Fall/2009. The student Apache Virtual Server configuration was replaced with an individual student, VMWare-virtualized, LAMP server. This permitted several advantages: a) each student was provided root access to their individual Linux server and could experiment and alter LAMP configuration files, e.g., httpd.conf, my.cfg and php.ini, b) the student VMWare virtual server could be used for other current or future CIS department courses, e.g., Linux System Administration, c) remote access and web server browsing can be accessed by either IP address or DNS domain name, and d) the conceptual introduction of server virtualization into the course content. Since private IP addresses were mapped to public IP addresses, the IT department was required to properly configure the university routers, switches and firewalls

The change in the technological infrastructure to a VMware virtualization provided significant improvements and increased flexibility in course administration, e.g., virtual server clones, student isolation, and security. It is also important to note that improvements in retention and performance associated with the third section offering occurred in a nine-week course length rather than a fifteen-week course. While not an objective of this paper, the differences between online, on-ground, and hybrid instructional formats in relationship to program language curriculum needs to be investigated in context of a LAMP platform.

2. LESSONS LEARNED

Every curriculum change is accompanied by both challenges and opportunities. The primary challenge of integrating LAMP in this study seems to have nothing to do with content, but with the instructional delivery method chosen. It was originally decided that, "what other course would be better to offered in online instructional delivery method than a LAMP web application development course?" The results of this case study indicates that there was significant improvement in student performance and retention when the course was changed from an online format to a hybrid format, accompanied by instructor lectures, demonstrations, and lab time.

LAMP includes a significant programming language component, PHP. While PHP may be easy-to-learn, it may face the same instructional challenges as other programming languages. Student background data was not analyzed in this study and may be an intervening factor.

In a previously cited case study concerning the use an Open Source Java-based web development platform, Chung & McClane (2002) cited the amount of extra work required by instructors. While some text books in LAMP do exist, there was considerable amount of extra work required by the instructor to develop instructional materials, student documentation, and evaluation instruments for this LAMP course. LAMP is not a mainstream curricular topic like WISA, and hence, instructional support materials are limited.

While no empirical analysis was conducted, it was observed that student satisfaction seemed to be high when students completed their projects. It was concluded that students appreciated their success because they could bet-

ter relate their in-class experiences to real-world dynamic, database-driven web sites.

The most significant opportunity and success of this case study was the technological implementation of LAMP using a VMWare Virtual server. While the low cost of Open Source software is well-known, scheduling, installing, and maintaining lab resources is a significant challenge. A \$10,000 initial investment in a VMWare server originally permitted up to forty virtual student LAMP servers that could be accessed remotely by students and faculty.

The cooperation and coordination between the Robert Morris University's Computer Information Systems and Information Technology departments to set up the VMWare server was excellent. Creating and maintaining student virtual machines was the responsibility of the faculty member. While there was additional work required by the instructor to support the virtual infrastructure, it was significantly less than previous experiences of maintaining a physical lab environment. Furthermore, the benefits of applying virtualization to other IS courses which also required physical lab support not discussed, substantially exceeded the virtualization benefits of this LAMP case study.

3. SUMMARY AND CONCLUSION

Web development and programming have been included in the IS curriculum for many years. It may be desirable to view the IS web curriculum from an integrated platform perspective, rather than separate isolated individual courses. WISA (a Microsoft Windows-based platform) and LAMP (an open source-based platform) are two of many dynamic, database-driven web platforms used by industry. The ease-of-use of Windows-based dynamic web platforms may be a significant reason why many IS curricula embrace the Windows solution.

This paper reviewed the case history of adding a project-based LAMP course to an IS curriculum that is already Windows-centric. With a minimum of course prerequisites, students were introduced to an open source alternative to teach dynamic, database-driven web developments using Linux, Apache, MySQL and PHP. Preliminary data indicates that this type of LAMP course may not a good candidate for on-line delivery.

The focus of this paper was not to determine which dynamic web development platform was

the best or should LAMP replace WISA. Rather, the conclusion of this paper indicates LAMP may complement any web development platform in use for a given IS curriculum, add to open source awareness, and provide IS students with a project experience with a minimum of course prerequisites.

The focus of this paper was to increase awareness of LAMP technologies within the IS curriculum. Further study is needed to determine the importance for LAMP technologies within industry and the IS curriculum. It is also recommended that additional study be conducted to determine the importance of the concept of "virtualization" as a content item in the IS Model Curriculum, as well as, its role of virtualization in supporting the IS curriculum.

4. REFERENCES

- Ballard, P., & Moncur, M. (2009). *Sams Teach Yourself Ajax, Java Script and PHP. Apache*. Sam's Publishing, Indianapolis.
- Cecchet, E., Chanda, A. Elnikety, S., Marguerite, J. & Zwaenepoel, W. A Comparison of Software Architectures for E-business Applications. Retrieved June 9, 2010 from [http://74.125.155.132/scholar?q=cache:l8rh4lucfM0J:scholar.google.com/+A+Comparison+of+Software+Architectures+for+E-usi-ness+applications&hl=en&as_sdt=800000000000](http://74.125.155.132/scholar?q=cache:l8rh4lucfM0J:scholar.google.com/+A+Comparison+of+Software+Architectures+for+E-usi-ness+applications&hl=en&as_sdt=80000000000)
- Converse, T. & Park, J. (2004). *PHP5 and MySQL Bible*. Wiley, Indianapolis.
- Cholakov, N. (2008). On Some Drawbacks of the PHP Platform. *Proceedings of the 9th International Conference on Computer Systems and Technologies*.
- Chung, W. & McClane D. (2002). Developing and Enhancing a Client/Server Programming for Internet Applications Course. *Journal of Computing Sciences in Colleges*, 22(1), 79-91.
- Dionisio, J., Dickson, C., August, S., Dorin, P., & Toal, R. (2007). An Open Source Software Culture in the Undergraduate Computer Science Curriculum. *ACM SIGCSE Bulletin*, 39(2), 70-74.
- Harris, A. (2004). *PHP 5/MySQL Programming for the Absolute Beginner*. Thompson Course Technology PTR, Boston.
- Janicki, T., Kline, D., Gowan, J., & Konopaske, R. (2004). Matching Employer Needs With IS Curriculum: An Exploratory Study. *Information Systems Education Journal*. 2(21).
- Learn Computer. (2010). Is LAMP Still a Popular Choice for Web Development? Retrieved June 9, 2010 from <http://www.learncomputer.com/lamp-popular-choice/>
- Lecky-Thompson, E., Eide-Goodman, H, Nowicki, S. & Cove, A. (2005). *Professional PHP5*. Wiley, Indianapolis.
- Lecky-Thompson, G. (2008). *Just Enough Web Programming with XHTML, PHP and MySQL*. . Thompson Course Technology PTR , Boston.
- Misfit Greek (2010). PHP versus ASP.NET – Windows versus Linux – Who's the fastest? Retrieved June 9, 2010 from <http://misfitgeek.com/blog/aspnet/php-versus-asp-net-ndash-windows-versus-linux-ndash-who-rsquo-s-the-fastest/>
- Meloni, J. (2008). *Sams Teach Yourself PHP, MySQL and Apache*. Sam's Publishing , Indianapolis.
- My Solutions. (2009) Advantages and disadvantages of PHP. Retrieved June 9, 2010 from <http://findmysolution.blogspot.com/2009/07/advantages-and-disadvantages-of-php.html>
- Open Source Initiative, (2010). Definition of Open Source. Retrieved June 14, 2010 from <http://www.opensource.org/>
- MySQL Enterprise Server 5.1 (2010). Retrieved June 2, 2010 from <http://mysql.com/products/enterprise/server.html>
- Perry, A. (2003). A Comparison of two major dynamic web platforms (LAMP vs. WISA). Retrieved June 9, 2010 from <http://www.shawnolson.net/a/302/a-comparison-of-two-major-dynamic-web-platforms-lamp-vs-wisa.html>
- PHP.net. (2010). Web Services. Retrieved June 14, 2010 from <http://php.net/manual/en/refs.webservice.php>
- PHP Usage Statistics (2010). Retrieved June 9, 2010 from

<http://trends.builtwith.com/framework/PHP>

Sheldon, R., and Moes, G. (2005). *Beginning MySQL*. Wiley , Indianapolis.

Shifflet, C. (2005). *Essential PHP Security Handbook*. O'Reilly Media Incorporated.

Wang, X. (2009). A Practical Way to Teach Web Programming in Computer Science. *Journal of Computing Sciences in Colleges*, 22(1), 211-220.

Web Master Tips. (2006). Microsoft Hosting vs. Linux Hosting. Retrieved June 14, 2010 from <http://tips.webdesign10.com/microsoft-hosting-vs-linux-hosting>

Appendix

Course Content in Relationship to Other Courses in the Curriculum (Table 1)	
Open Source eCommerce Web Development (LAMP) Course Content	Related Courses and Sample Course Content
Putty and WinSCP Introductory Linux commands and fundamental administration concepts, e.g., cat, ls, pwd, cd , mkdir, chmod, etc.	Linux System Administration – Introductory and intermediate-level Linux commands and administrative concepts, e.g., Linux file systems, processes, system initialization, shell programming, etc.
Fundamental relational database concepts, My SQL data definition (CREATE TABLE and INDEX) and data manipulation (INSERT, UPDATE, DELETE and SELECT) commands, and security concepts	Database Management Systems - Data design and modeling, normalization, integrity concepts, introductory and intermediate-level Oracle SQL data definition and data manipulation commands, user administration and security, and introduction to PL/SQL
Review of (X)HTML tables and forms, CSS, and introductory JavaScript	Introduction to Web Development and eCommerce Technologies
Apache Administration Concepts and Security, httpd.conf, Apache user authentication, permissions and access control. MySQL with Apache user authentication	Linux System Administration – Introductory Apache administration concepts.