

Design and Delivery of Multiple Server-Side Computer Languages Course

Shouhong Wang

Charlton College of Business
University of Massachusetts Dartmouth
Dartmouth, MA 02747-2300
USA
swang@umassd.edu

Hai Wang

Sobey School of Business
Saint Mary's University
Halifax, NS B3H 2W3
Canada
hwang@smu.ca

Abstract

Given the emergence of service-oriented architecture, IS students need to be knowledgeable of multiple server-side computer programming languages to be able to meet the needs of the job market. This paper outlines the pedagogy of an innovative course of multiple server-side computer languages for the undergraduate IS majors. The paper discusses the rationale of why the proposed pedagogy is different from and improves the traditional methods. The paper provides a description of the approach to teaching a multiple server-side computer languages course. Based on our experiences in the past years, it is concluded that a single course of multiple server-side computer languages is useful and feasible for the IS programs.

Keywords: .NET, application development, client-server, computer programming, computing skills, creative problem solving, Extensible markup language (XML), instructional pedagogy, job skills, programming languages, Service oriented architecture (SOA), Web design and development.

1. Introduction

There have been critical discussions on the IS curriculum design during the last several years (Topi et al. 2010). The most notable trend in the IS curriculum renewal movement is to develop more new IS electives to meet the needs of the job market of IS graduates (Drinka and Yen 2008). This need in the job market has considerable implications for the designing of IS elective courses in the educating of the next generation of IS professionals. IS students must acquire the fundamental theories of IS as well as the essential practical skills of the computer applications, while developing the life-long learning ability in information technology during their IS education. (Surendra and Denton 2009; Topi et al. 2010). Technical skills should focus more on problem solving and practical applications (Downey et al. 2009). The IS curricula have changed over the past years to meet the requirements of the job market as well as the requirements of the accreditation organizations such as AACSB and ABET. Nevertheless, programming remains a core requirement in most IS

programs, which is one of the most difficult subjects in the IS curriculum (Sleeman 1986; Malik 2000; Palma 2001; Robins et al. 2003; May and Dhillion 2009). This paper reports how this challenge is met by designing the contents of a course of multiple server-side computer programming languages for undergraduate IS majors.

2. Overview of the Course

In the modern service-oriented age, the development and maintenance of Web based applications still relies heavily on applications of third generation computer languages regardless of the advances in fourth generation computer languages and a variety of software packages. To meet the challenges of the ever-changing information technologies, educators need to offer courses of important server-side programming languages for their undergraduate IS majors. On the other hand, undergraduate IS majors cannot afford to learn multiple server-side computer languages on the one-language-one-course basis. The key to the solution to this problem is to

make a pedagogical paradigm shift and to develop a course of multiple server-side computer languages.

Few guidelines for IS elective courses of server-side computer programming can be found in the literature or on the Internet. The selection of server-side languages for a course is a crucial task for the pedagogy design. The design components of such a course are based on four considerations. First, the selected server-side languages must be commonly used in the industry. Second, the selected server-side languages must be representative, and should cover essential features of all kinds of server-side languages. Third, the selected server-side languages do not require additional computing resources in our computing lab. Fourth, the total workload for students should be manageable. Taking these factors into account, three major server-side computer programming languages were selected for our course. These languages are ASP.NET (with VB.NET), PHP, and XML (with XSLT, DTD and XML Schema).

Due to time constraints, it is impossible for students to learn all these server-side languages in great detail. Nevertheless, students in this course are expected to have general knowledge of server-side languages as well as to be able to develop basic skills of programming. The central methodology applied to this course is the learning of languages through typical examples. Specifically, we teach typical problems of Web applications and their solutions through the use of these computer languages.

The course described in this paper is entitled "Web Application Development and Programming," and is designed as an elective course for junior IS majors who are pursuing the careers as application developers or Web content managers. The prerequisite of this elective course is an introductory computer programming course, which provides an introduction to client-side computer languages. Although the contents of the introductory programming course depend upon the individual instructor, the course normally covers HTML, JavaScript, and VB.NET. The elective course described in this paper is taught over one semester, normally 3 credit hours over 14 weeks. In its design, this course consists of two distinct modules. The Teaching Module provides an overview of representative server-side computer languages in service-oriented architectures. The Project Module provides an opportunity to apply the server-side computer languages involving hands-on projects.

3. Justification of the Course

The structure of the 2010 IS model curriculum (Topi et al. 2010) includes two parts: core IS courses and elective IS courses. The core IS courses include Foundations of IS; Enterprise Architecture; IS Strategy, Management, and Acquisition; Data and Information Management; Systems Analysis and Design, IT Infrastructure; and IT Project Management. The elective IS courses include Application Development; Business Process Management; Collaborative Computing; Data Mining/Business Intelligence; Enterprise undergraduate IS majors. In the service-oriented architectures, the client side is responsible for data presentation and data

Systems; Human-Computer Interaction; Information Search and Retrieval; IT Auditing and Controls; IT Security and Risk Management; Knowledge Management; and Social Informatics. The 2010 curriculum recommendation clearly states that, unlike its predecessor model curricula, it does not provide specific courses that address the IS foundational skills and knowledge or domain-specific skills and knowledge. The design and delivery of IS courses, especially electives, for an IS program depend upon the computing environment and its specific needs.

Web services, software-as-a-service, and cloud computing are all important elements in the modern service-oriented architectures of computer-based systems. The present server-side programming course provides the necessary body of knowledge for the current trends. In terms of the 2010 IS Model Curriculum (Topi et al. 2010, Figure 6), the proposed course fits "Application Development" which is on the top of the list of elective IS courses and is relevant to all career tracks. In our IS programs, e-commerce is a concentration area. Our experiences in the past several years have indicated that this course is an important component of the e-commerce concentration, and meets various career tracks which are suggested in the 2010 curriculum recommendation. During the past three years, the IS students' demands for this elective course have been increasing significantly although the overall enrollment of IS majors remains steadily low.

3.1. Need for learning multiple server-side computer languages

Due to historical and technical reasons, there have been many server-side computer languages. The need for learning multiple computer languages can be perceived from a variety of views. For instance, Perl is still important since many legacy Web-based systems are Perl-based. However, modern large service-oriented systems heavily rely on Java and .NET. On the other hand, PHP is widely used in many small scale service-oriented applications. Our argument is that, given the variety of client-server middleware in the computer-based systems, undergraduate IS majors need to learn the distinct features of the major server-side languages.

3.2. Sufficient coverage of important concepts of server-side programming

The course shall cover the key concepts of server-side computing that supports interaction between the clients and server as well as the actions on the server. The key concepts that are particularly relevant to service-oriented architectures are identified and are summarized as follows.

3.2.1. Job division in service-oriented architectures

The architecture of web-based applications covers a broad range of subjects, including information orientation, communication orientation, and service orientation. In the present study, the focal point is placed on the service-oriented architecture. Job division between the client and server in the service-oriented architectures is an essential concept for input, and the server side is responsible for data processing, as illustrated in Figure 1.

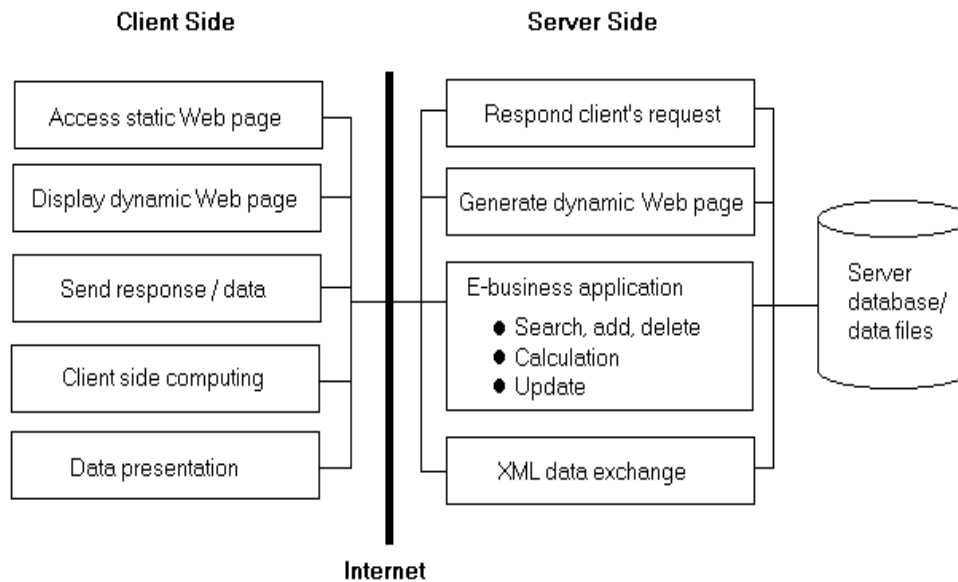


Figure 1. Client-Server Job Division in Service-Oriented Architectures

3.2.2. The mechanism of service-oriented process

Service orientation is implemented through interactions between the client and the server. Client-side computing programs, Web browser, the Internet protocol, server-side programs, and server database or data files comprise the chain of service-oriented processes. The ultimate service-oriented processes on the server side are recording, updating, and searching of data on the server in response to the requests from the client side.

3.2.3. Dynamic Web pages generated by server-side programs

One of the key tasks of server-side programs is the generation of dynamic Web pages, no matter what server-side computer language is used to implement service-orientation. Dynamic Web pages implement contingent actions of the server in response to the client's specific requests.

3.2.4. Uniform format of data exchange

XML, the uniform data format for databases, has become a necessary component in service-oriented architectures. XML and its companion languages (XSLT, XML Schema, etc.) make data accurate in exchanging, efficient in searching, and independent of presentation.

In summary of the above discussion, there is little doubt that knowledge of various server-side computer languages are substantial, and can contribute to the career development for undergraduate IS majors. Now, the question is how we can teach these multiple server-side languages in a feasible way.

3.3. Feasibility of teaching multiple server-side languages

It is impossible to teach multiple languages by using the traditional one-language-one-course approach. To compact material of multiple languages into a single course, we must change the traditional approach to teaching computer

languages that begins with syntax and ends with disjointed examples of algorithms. Typically, we use simple examples of client-server end-to-end interaction to teach such a course. We take advantage of synonyms; that is, despite the variety of syntax in the different languages, languages share many common key concepts. Once a key concept is introduced in one language, it should be understandable for students when learning another language.

3.4. The course adds more value to student learning

This innovative course is fast-paced, and encourages self-learning on the students' part. Since the nature of this course is practice-oriented, multiple small-scale projects will be required. The central point of the shift from one-language-one-course to multiple-languages-one-course is to add more value to student learning by giving students discipline in self-learning and encouraging them to apply learned knowledge to the real world. According to our observations, students feel stress in the same way as in any other programming course; but, the never felt bored in this course.

4. Teaching Module

It was difficult to find a single integrated textbook that meets our course needs. To give students a comprehensive guideline for their studies, lecture notes have been developed for this course during the past years. These lecture notes have been revised many times, and recently have been included in a textbook of computer programming languages for IS students (reference is available upon request).

The Teaching Module is divided into three units based on the selected server-side languages. Each unit has a written exam. A sample course syllabus is exhibited in Appendix I.

The features of the course competency, relevant paradigms, and service-oriented applications are presented below.

4.1. Unit 1: ASP.NET

Compared with the Java platform, ASP.NET may not be as efficient as Java in terms of execution performance, but is a top popular server-side language due to the popularity of the Windows platform. In our experience of teaching server-side languages, ASP.NET is particularly superior in a computing environment in providing a seamless connection with databases (MS SQL Server or Access).

An ASP.NET program has two parts: the server processing logic and the Web form. We use VB.NET for the part of user interface logic. Although the prerequisite course of this course includes VB.NET, we still use the first two or three classes for the reviewing VB.NET.

The major components of this unit include the structure of ASP.NET program, HTML controls and Web controls, code-behind programming framework, and reading and writing database or files on the server.

A great feature of ASP.NET is the seamless connection with Access databases through ADO.NET. One interesting part of this unit is the use of SQL for Web applications. Although most students in this course have not taken a database course, they seem to be able to deal with simple SQL embedded in ASP.NET. Typical client-server end-to-end examples for this unit include online order processing and student database searching.

We use 40% of the class time for this unit. Our computing lab is equipped with PCs using the Windows platform. The computing environment for this unit is Microsoft Visual Studio 2010.

4.2. Unit 2: PHP

To meet the challenge of information technology in the service-oriented world based on long-term considerations, we recognize that general knowledge of PHP will be an asset for undergraduate IS majors since PHP is free open source software and is widely used in service-oriented applications. In terms of syntax, PHP is very similar to Perl which is considered to be old but is still running on many servers.

The focus of this unit is placed on the structural difference between ASP.NET and PHP. Unlike ASP.NET, PHP does not have a division in the server processing logic and the Web form; instead, a PHP program mixes both server-side data processing and printing (or generating) a dynamic Web page for the client.

Similar to teaching ASP.NET, the major features of PHP, including commonly used commands, reading/writing data files on the server, and data relay through multiple forms, are explained using examples. Typical client-server end-to-end examples for this unit include online order processing and airticket reservation.

The computing environment for this unit is EasyPHP (2011), an open source software PHP environment. EasyPHP is easy to install and easy to use. Students take the responsibility to install EasyPHP on their computers. PHP with MySQL is a good combination for delivering the concepts of server-side programming. We use 25% of the class time for this unit. Our observations have shown that, in comparison with ASP.NET, PHP has less built-in features and is easier to learn when students have learned ASP.NET.

Currently, we are using the Windows platform for the PHP unit. It will certainly be a new direction for this unit if in the future we are going to integrate PHP and databases on Linux servers which are widely used in industry client-server systems.

4.3. Unit 3: XML

XML has been widely applied in Web applications, and the application is an important body of knowledge for undergraduate IS majors. Although the concept of XML is not difficult to learn, students often have difficulties in this unit. The problems with this unit are not with XML itself, but with the confusion caused by so many XML companion languages.

The major components of this unit include features of XML instance documents, data structure of an XML document, Cascading Style Sheets (CSS), Extensible Style Language (XSL), Document Type Definition (DTD), XML Schemas, and XML document validation. Typical client-server end-to-end examples for this unit are the use of a single data set in the XML format for different purposes in separate service-oriented applications.

This unit includes XHTML and a very brief overview of XBRL. However, these components are optional.

For the time being, there is little direct connection between XML and ASP.NET or PHP. It would be interesting to watch closely how the IT industry establishes application linkages between XML and ASP.NET or PHP.

We use 35% of the class time for this unit. The computing environment for this unit is Microsoft Internet Explorer. Free user-friendly XML validators such as CoreFiling (2011) are also used for this unit.

4.4. Examinations

After each of the three units, students are required to take an exam. Students are presented with uncompleted computer programs with numerous blanks. Students are required to complete the programs, explain the purposes of these programs, and sketch the expected execution results of the programs. The designs of the tests strongly suggest that the global aspects of service-oriented applications, including context, client-server interaction, data processing on the server, and service-oriented process outcomes, are more important than the local aspects (syntax) to learn for undergraduate IS majors.

5. Project Module

The Project Module of the course concentrates much more on "action learning", and requires students to conduct their quasi-real-world projects. In this module, students were encouraged to work in small teams, typically, of two people. Nevertheless, many students chose to work on projects without joining team. Each group (or individual) applies the server-side languages discussed in the Teaching Module for their three projects. It is expected that, upon the completion of this course, these students are able to write programs in these computer languages to solve simple problems related to service-oriented applications.

5.1. Project Configuration

After each unit presented in the Teaching Module, students form project teams if they choose to do so to share common

expectations of learning. Three stages can be outlined for each project. In the first stage, students choose project topics based on their general understanding of the server-side languages and their cases. In the second stage, intensive programming activities are undertaken for each project. In the final stage, the project product is formalized in a report. Although the components of reports may differ significantly from project to project, the report structures adopted in three projects are almost the same. The first part of the report will describe the background of the business problem to be solved. The second part of the report provides a technical description of the project including source code. The third part of the report provides screenshot examples of program execution results.

5.2. Project Requirements

We clearly set criteria for each project for assessment, as briefly described below.

5.2.1. Minimum requirements for ASP.NET projects

Possible project topics include online shopping, online survey, online billing, online payment, etc. The minimum requirements for ASP.NET include: An HTML home page that starts an ASP.NET program, followed by two times of interaction between the client and server implemented by ASP.NET; At least 1 data file (.txt file) and 1 database table (MS Access) used for the programs for data storage and search.

This course emphasizes on the interaction between client and server, but de-emphasizes the static Web page itself (such as hyperlinks, client-side calculations and image manipulations through JavaScript).

A project report includes:

- Description of the project;
- Source code of the home page and all programs;
- The screenshots that illustrate the interaction between client and server; and
- Data files and MS Access database that will illustrate the data storage on the server, and sample data for the project.

5.2.2. Minimum requirements for PHP projects

The requirements for PHP projects are similar to the requirements for ASP.NET, except for the database which is not required for PHP, because the setting database connections for PHP is beyond the scope of the server-side language course. Nevertheless, many students were able to set MySQL or SQL Server for their projects by themselves through learning from the Internet.

5.2.3. Minimum requirements for XML projects

The minimum requirements for XML projects are one XML document, its Schema and validation, and two XSLT programs for the XML document for two distinct applications.

A project report should include:

- Description of the project;
- Source code of all programs (XML, XSLT, Schema);
- Tree diagram for the XML document;
- Validation of the XML document against its schema;
- The screen shots of the two distinct applications.

Students are asked to follow the structures of the service-oriented application examples they learned in the Teaching Module and to add their own creative components for their

projects. For instance, after learning the typical online ordering process, students are generally able to write ASP.NET and PHP programs for their own projects such as hotel reservation, online auction, etc. Based on our observations, we are convinced that students are able to write complete programs from scratch in each of the languages we taught for typical simple service-oriented application projects. Computer literacy for undergraduate IS majors means "be knowledgeable with a variety of computer languages", and such an aptitude can be developed only through hands-on practices. (Woszczyński et al. 2005; Kung et al. 2006).

6. Findings and Conclusion

This server-side language course has been offered in our IS programs at the two universities of the co-authors during the past six years. Although the teaching evaluation questions at the two universities were not same, the format and major aspects were similar and comparable. Only the questions of overall satisfaction with the course and free-format comments were considered and analyzed for this study. Questions were rated on a five-point scale for students to answer. A four-point or higher mark for a question was considered a positive answer to the question. Eighty seven (87) filled course evaluation forms from this course have been recorded for this study. According to the course evaluations, all students gave a positive answer for the question of overall satisfaction with the course. According to our observations of the exam results and projects as well as the comments from the students, students who were eligible to take this course have no difficulty in learning the techniques outlined in this paper and conducting projects. After the course, some students have demonstrated their ability to participate more formal service-oriented application projects. We have compelling examples of follow up comments by those who took this course. In fact, we have collected several comments from the students who are working in the Web application development field subsequent to taking this course, and have presented them to our current students (see Appendix II). Although our sample size was relatively small, we are convinced that knowledge of multiple server-side languages will certainly be beneficial for the IS programs.

Overall, the results indicate that the course discussed in this paper is a valuable component of the renewal curriculum for IS programs. We believe that a single compact course of multiple server-side languages for undergraduate IS majors is useful as well as feasible. All computing resources (hardware and software) needed for the implementation of this course are commonly available at all educational institutions. Thus, it is certain that our success can be replicated in other IS programs.

One of the most important aspects of effective IS education is to help IS students to develop problem solving skills to meet the challenges of the fast changing IS field. IS educators need to have a greater understanding of problem solving schemes in order to design innovative curricula that will emphasize students' practical skill sets. Our study has made contributions to innovative education as it has initiated and implemented a course of three major server-side programming languages for the IS majors. The summarized pedagogy of this course can be applied by others to change the traditional methods of teaching computer programming languages by shifting from single-emphasis to variety-

emphasis. The most compelling implication of this study for the IS education is the recognizing of the potentially positive effect of pedagogical re-design in enhancing the students' problem solving skill sets.

References

- CoreFiling. 2011. XML Schema Validator <<http://tools.decisionsoft.com/schemaValidate/>>, [accessed August 3, 2011].
- Downey, J. P., McGaughey, R., and Roach, D. 2009. "MIS versus computer science: an empirical comparison of the influences on the students' choice for major," *Journal of Information Systems Education*, 20(3), 357-368.
- Drinka, D. and Yen, M. Y. 2008. "Controlling curriculum redesign with a process improvement model," *Journal of Information Systems Education*, 19(3), 331-342.
- EasyPHP. 2011. Open source software PHP <<http://www.easypHP.org>>, [accessed August 3, 2011].
- Guthrie, C. 2010. "Towards greater learner control: Web supported project-based learning," *Journal of Information Systems Education*, 21(1), 121-130.
- Kung, M., Yang, S. C., and Zhang, Y. 2006. "The changing information systems (IS) curriculum: a survey of undergraduate programs in the United States," *Journal of Education for Business*, 81(6), 291-299.
- Malik, M. A. 2000. "On the perils of programming," *Communications of the ACM*, 43(12), 95-97.
- May, J., and Dhillon, G. 2009. "Interpreting beyond syntactics: a semiotic learning model for computer programming languages," *Journal of Information Systems Education*, 20(4), 431-438.
- Palma, P. D. 2001, "Why women avoid computer science," *Communication of the ACM*, 44(6), 27-29.
- Robins, A., Rountree, J., and Rountree, N. 2003. "Learning and teaching programming: a review and discussion," *Computer Science Education*, 13(2), 137-172.
- Sleeman, D. 1986. "The challenges of teaching computer programming," *Communication of the ACM*, 29(9), 840-841.
- Surendra, N. C., and Denton, J. W. 2009. "Designing IS curricula for practical relevance: applying baseball's 'Moneyball' theory," *Journal of Information Systems Education*, 20(1), 77-85.
- Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F. Jr., Sipior, J. C., and Vreede, G., J. d. (2010). "IS 2010: Curriculum guidelines for undergraduate degree programs in information systems," *Communications of the Association for Information Systems*, 26, Article 18.
- Woszczynski, A. B., Guthrie, T. C., and Shade, S. 2005. "Personality and programming," *Journal of Information Systems Education*, 16(3), 293-299.

Author Biographies

Shouhong Wang is a Professor of Management Information Systems at University of Massachusetts Dartmouth. He received his PhD in Information Systems from McMaster University. His teaching and research interests include computer programming, knowledge management, e-commerce, and business intelligence. He has published over 100 papers in academic journals and several books in MIS.



Hai Wang is an Associate Professor of Computing and Information Systems at Saint Mary's University. He received his PhD from University of Toronto. He has published over 30 papers in academic journals and a textbook of multiple computer programming language. His teaching and research interests are in areas of computer programming, knowledge management, e-commerce, and data warehouse and data mining.



Appendix I. Sample Course Syllabus of the Course

Course Title: Web-Based Application Development: Server-Side Programming

Prerequisite(s): Introduction to Programming and Problem Solving

1. Course Description:

This MIS elective course will help students to develop the ability of using computer programming to develop Web-based computing applications. Students will use advanced programming languages including VB.NET, ASP.NET, ADO.NET, PHP, and XML to implement Web-based business applications. This course covers physical design, programming, testing, and implementation of the Web-based information system. The emphasis of this course is placed on server-side programming and implementations of object-oriented, client-server designs using integrated Web application development environments.

2. Course Objectives:

Upon completion of this course students will:

- a. Be able to develop Web-based 3-tiers (Web client, Web server, and application server or the legacy system) computing applications by using modular design, programming, and software testing and validation techniques.
- b. Be able to develop implementation standards that are used rigorously as project teams complete a significant system, and to develop a conversion and training plan for the system.

3. Competencies and Contact Hours:

Competencies	Contact Hours	
The student will be introduced to:		
• Web-based business applications	3	3
• Tier structure of Web-based applications	3	6
The student will understand:		
• The development cycle of server-side software	6	12
• Programming tools for server-side software	12	24
The student will be able to:		
• Write VB.NET, ASP.NET, ADO.NET, PHP, and XML to develop Web-based applications	12	36
• Test server-side programs	6	42
• Document the developed server-side programs	3	45

4. Course Material:

(1) Required Textbook (the co-authors' textbook on computer programming languages for business).

(2) Software for HTML, JavaScript, VB.NET, ASP.NET, ADO.NET, MS Access, XML:

Notepad, MS Internet Explorer, Microsoft Visual Studio .NET 2010, Microsoft Office 2007. They are all available in the Lab.

(3) Software for PHP:

Open source software EasyPHP. Students have the responsibility to download it from <<http://www.easyphp.org>> and install it on your own laptop, following the instructions in the textbook. EasyPHP might be available in the library lab.

5. Classroom Policy:

(1) Students are required to take three tests on .NET, PHP and XML. Tests are open-book and closed-computer.

(2) Students are expected to conduct three Web application projects that involve the use of the computer languages learned from this course. Each student has the responsibility to install open source PHP environment EasyPHP for the PHP project.

6. Evaluation Policy:

Tests (ASP.NET, PHP and XML)		45pt
Three Project (ASP.NET, PHP, XML)	(25pt+12pt+18pt)	55pt
Report presentation	(2pt 1pt 4pt)	
Usage of programming languages	(18pt 8pt 12pt)	
Demonstration	(5pt 3pt 2pt)	

100pt

7. Schedule of Weekly Activities (subject to change depending on progress):

Week	Schedule Description	Important Dates
1	Course Introduction Refresh HTML, JavaScript, and VB.NET (MIS 212) Introduction to server-side programming (Java, CGI Perl, ASP.NET, PHP)	
2	Introduction to ASP.NET The Structure of an ASP.NET Program	
3	HTML Controls vs. Web Controls HTML Controls: Submit button, Textbox Checkbox, Radio Button, Select	
4	Web Controls Validation Controls The Code-Behind Programming Framework	
5	ASP.NET Web Page Application Examples Sending Email Calendar File Input/Output Security	
6	ADO.NET, database, and SQL in Web applications Project overview Review ASP.NET	
7	Introduction to PHP The Structure of a PHP Script Web Page to Trigger PHP	Test for ASP.NET ASP.NET Project Due
8	PHP Functions, if-else statement Read Data Files from the Server while loop	
9	Write Data Files to the Server Relay Data through Multiple Forms Using Hidden Fields PHP project overview	
10	Introduction to XML Feature of XML Instance Documents Declaration, Tags and element, Attribute, Comment line and editorial style, Empty tag Cascading Style Sheets (CSS)	Test for PHP
11	Extensible Style Language (XSL) CSS vs. XSL Document Type Definition and Validity Simple XML Examples with Internal and External DTD Features of DTD	PHP Project Due
12	XML Schemas Schema Elements Data Element, Element Name, and Element Type	
13	Business Applications of XML	
14	XHTML XBRL Review XML	Test for XML

15	Wrap up.	XML Project Due
----	----------	------------------------

8. Project Requirements

Each student must submit a project report for each project. Minimum requirements are:

(1) **ASP.NET**: An HTML home page that starts an ASP.NET program, followed by at least two times of interaction between the client and server implemented by ASP.NET. At least 1 data file (.txt file) and 1 database table (MS Access) used for the programs for data storage and search. Possible application topics include password verification, online shopping, online survey, online billing, online payment, etc. Note that this course emphasizes on the interaction between client and server, but de-emphasizes static Web page itself (such as hyperlinks, client-side calculations and image manipulations through JavaScript).

Your report should include:

- Description of your project
- Source code of the home page and all programs
- The screen shots that illustrate the interaction between client and server
- Data (.txt files and MS Access database) that illustrated the data storage on the server, and sample data for project

(2) **PHP**: Similar to the requirements for ASP.NET.

(3) **XML**: 1 XML document and its tree diagram with 3 levels, at least 10 nodes of elements and attributes; Schema for the XML document and validation; Two XSLT documents for the XML document for two applications.

Your report should include:

- Description of your project
 - Source code of all documents (XML, XSLT, XML Schema)
 - Tree diagram of the XML data set
 - The screen shots of applications, and validation (you may access CoreFiling.com for XML Schema Validator <<http://tools.decisionsoft.com/schemaValidate/>>)
- (**Note**: Test on XML covers DTD, and project covers XML Schema.)

9. Rubrics for Projects:

		Assessment Comments
		4 (Excellent) 3 (Very Good) 2 (Fair) 1 (Poor)
Report presentation	<ul style="list-style-type: none"> ◦ E-Business applications ◦ Clearly developed and linked introduction and application. ◦ Documentation 	
Usage of the languages	<ul style="list-style-type: none"> ◦ Error free programs ◦ Intensiveness of use of the language. ◦ Clear structure ◦ Professional programming style 	
Demonstration	<ul style="list-style-type: none"> ◦ Interesting ◦ Impressive output ◦ User friendly 	

Appendix II. Samples of Students' Feedback Comments on the Course

Former Student #1:

Hello, Professor:

I think it was a great help in getting my current job as a Web Developer. I recently finished building <http://www.xxxx.com/> by myself using drupal. My next project is to rebuild the entire software suite that they sell to local municipalities, etc. This software suite is enormous in scale so it is a rather scary thing knowing what is ahead of me. I will be building the software website using visual studio 2008, asp.net(vb.net), MS SQL Server 2008, and IIS 7.5. One of the things that I noticed at work and during my job searching is that mostly everyone today is using SQL Server or some other database software and visual studio. My company currently uses Access and SQL Server depending on which release the customer has. So I am gaining experience in both platforms. My main reason for touching base with you was just to thank you for everything but also to just let me know that it might be a good thing for your future students to learn some SQL Server and MySQL. Mainly it is important to be used to the Visual Studio environment. That has been the biggest adjustment I have had to make since joining this company. Learning Visual Studio and database interaction with SQL Server has been by far the most difficult part of the job. Again, thank you for everything you taught me, Professor.

A. D.

Former Student #2:

Hi, Professor:

As an Instructional Technologist, my tasks include testing, supporting and extending functionality to a variety of open source and commercial applications related to teaching and learning. In addition, I often must work with data and content encoded to meet industry standards. Exposure and understanding gained through MIS 312 impacts all of these functions. This is due to PHP, ASP .NET and XML being among the most prevalent languages used in industry today. Prior to my experience in MIS 312, I had not yet been exposed to ASP .NET programming. The experience gained through the ASP project enables me to better support applications currently being evaluated for deployment in this organization. XML is a major component in ensuring content interoperability for many modern learning management and media delivery platforms that I work with and evaluate. Because of the knowledge gained in MIS 312, I now have a much greater understanding of XML schema development. Even though I do work with PHP frequently, the freedom of the course project allowed me to experiment with relevant functionality I had not yet had the opportunity to work with professionally.

D. G.

Former Student #3:

Dear Professor,

I believe MIS 312 has greatly impacted my job hunting and work after the MIS program. I am currently a Programmer/Analyst for XXX Information Technology implementing Healthcare Information Systems for the XXX applications. Although XXX uses its own proprietary languages for its applications, it does use some of the web based technologies. For example, I was recently asked to write a web based version (HTML & XML) of a document for my group due to the fact that I have experience with those technologies. The experience I speak of is the experience I gained in the MIS 312 class. The class provided me with the skills needed to complete these types of tasks with ease and efficiency.

It is my opinion, through my experience, that these technologies are still being utilized. When hunting for technical positions, being well rounded in these technologies is an absolute plus. Even if it is not utilized at a particular company or organization I believe knowing the technologies shows you ability to understand the connection between client and server. It also shows you ability to learn new languages quickly and to be able to apply them for business purposes. Numerous positions I researched in my job hunting required experience in ASP.NET, PHP and XML.

Thanks,

R. D.



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©2011 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, editor@jise.org.

ISSN 1055-3096