Teaching Web Development Technologies:

Past, Present, and (Near) Future

Billy B. L. Lim
Applied Computer Science Department
Illinois State University
Normal, IL 61790-5150
**mailto:mbllim@ilstu.edu**

ABSTRACT

The advent of the World Wide Web not only fundamentally changes the way we do things, it also greatly changes how computer software are built. While this profound evolution of software development has caused many in the software industry to change their way of developing software, it has not caused many in the academics to change their way of teaching computing. This paper describes the evolution of a course that teaches web development technologies. The contents of the course in three different iterations, the observations made, the available educational resources employed, and the challenges ahead for the course are presented.

Keywords: Web Development Technologies, Pedagogical Pattern, Web Development Resources

1. INTRODUCTION

The field of information technology has experienced its shares of changes and evolution in the past few decades when compared to other disciplines. But none of the changes and evolutions in recent years is as profound as the introduction of the World Wide Web (WWW) in 1992 by Tim Berners-Lee at CERN, Switzerland.

In fact, to get a sense of the potential impact of the web on our society, consider Figure 1 given below [1]. The figure shows the web as it is compared to the inventions of electricity, radio, television, and telephone from 1900 to year 2000 (X-axis). As shown, the various ubiquitous household "technologies" have almost reached their full capacities (Y-axis) where as the potential for the web in only beginning to be explored. Needless to say, the growth of the web today is not as wild as during the heyday of the Internet explosion where millions of web pages were being created daily. Still, the web has been growing steadily. This is evident from many studies including a recent report from the FCC (Federal Communications Commission) where it is reported that web access jumped 158% in 2000 as compared to the year before and that 7.1 million homes and businesses had high-speed (i.e., DSL, cable modem, satellite, and wireless)
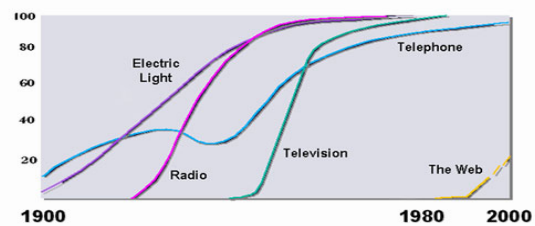
Figure 1: Comparisons of various inventions with respect to their capacities over time.

web access (CNNFn, '01).

Noting this explosive growth potential of the web, many businesses have already transformed their business models to adapt to and exploit this new world of opportunities. This includes reexamination of their business processes, system development architecture, personnel (re)training, etc. For the academics, information systems / computer science (IS/CS) departments need to also reexamine their curricula in order to prepare students to face the challenge of not only being able to function in a traditional data processing environment, but also be productive in a computing world that is now swamped with web technologies. But a recent search of the computing education literature shows that there are few publications that describe the study of web

development technologies, where all the major components of the web with respect to development are studied extensively. This is crucial to note, as the dire need to produce "web-enabled" graduates who can immediately fill the void in the industry becomes more and more apparent.

Some IS/CS departments have started incorporating web development technologies courses into their curricula and shared their findings (e.g., Lim 98, Walker and Browne 99) while others may have done the integration of these courses but simply did not report on their experiences. Yet others are not as fast to address the issue due to various reasons such as lack of resources and/or a model course to emulate. It is the latter group that this paper in intending to target.

This paper describes a course entitled *Web Development Technologies*, its evolution over the last few years, and how it can be integrated into IS/CS curricula. The objectives of the course are to study the major components of the web, to understand how they interrelate in system development architectures, and to develop web applications using the technologies studied. The course uses a breadth-first approach to introducing the pertinent technologies for web development. It is aimed at juniors, seniors, and graduate students who have, as prerequisites, two semesters of programming in C, C++, or Java and one semester of introduction to system analysis and design. When the course was offered the very first time a few years ago, it was offered as a *Special Topics in Systems Design* where students could take it as an elective in all the sequences offered by the department.

The Web Development Technologies course is now made permanent in the curriculum (ACS 353) and it serves as the anchor of the newly developed *Web Application Development* sequence. The sequence is available to students in the Bachelor of Information Systems major starting fall of 2001. The department also offers Bachelor of Applied Computer Science and Bachelor of Telecommunication Management degrees. The ACS 353 course is an elective course in these two majors presently.
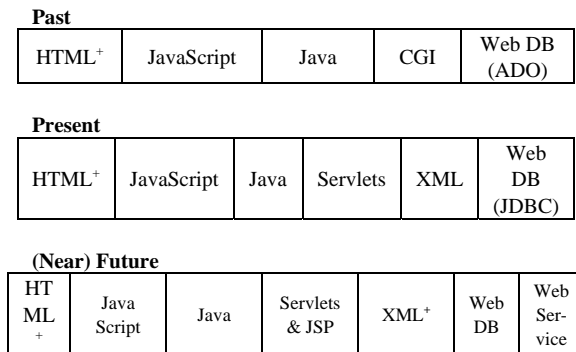
This intent of this paper is to describe the aforementioned course in its past, present, and (near) future forms. The contents of the course and the experience in delivering them, the various kinds of assignments given, the tools used, and this author's philosophy in teaching this course are the main focus of this paper.

The remainder of this paper is organized as follows. Section 2 describes the evolution of the course and the

educational resources that would be helpful for someone who might be interested in a similar offering. The course assignments and project are described in Section 3. A sample course outline is given in Section 4. Section 5 shares the author's experience in teaching the course and provides some observations. Finally, summary and concluding remarks are given in Section

2. TEACHING WEB TECHNOLOGIES: PAST, PRESENT, AND (NEAR) FUTURE

The following figure shows the topical coverage of the aforementioned Web Development Technologies course. The course spans over a semester of 16 weeks and the figures illustrates how the course has evolved and will continue to evolve in the span of just 4 years! The amounts of time spent on the topics are roughly depicted by the size of each of the boxes corresponding to the topics shown.

Past

$HTML^+$	JavaScript	Java	CGI	Web DB (ADO)

Present

$HTML^+$	JavaScript	Java	Servlets	XML	Web DB (JDBC)

(Near) Future

$HT ML^+$	Java Script	Java	Servlets & JSP	XML^+	Web DB	Web Service

Figure 2: The Evolution of the Web Development Technologies Course

2.1 Past

For the first few offerings of the course (which started in 1997), the beginning of the course was devoted to the coverage of introduction of the World Wide Web architecture, the HTTP protocol, and HTML as the language of the web (collectively denoted as $HTML^+$ in Figure 2). This was followed by the introduction of JavaScript as a client-side scripting language. Java was the natural choice as far as the next web-related technology is concerned since it was (and continues to be) prevalently used for a plethora of web applications. It inherent support for the web and networking applications was the emphasis of this component of the course. The next component of the course was CGI (Common Gateway Interface) using Visual Basic (VB) as the language of implementation. Perl, a popular CGI language, was not used due to student background (most of them have VB before coming into the course). The last component of the course was on accessing databases from the web. Here, ADO (ActiveX Data Object) via VB was the

technology used. For details of this "Past" version of the course, refer to (Lim 98).

2.2 Current
In the current offering of the course, note that the HTML$^+$ component is reduced somewhat in terms of coverage time. This is due to the fact that most students coming into the course have had some exposure to the language HTML and thus very minimal amount of time is spent on the language issue. Most of the introductory segment focuses on the HTTP protocol and the architecture of the web and web development (especially the N-tier architecture). This segment also lays down the foundation for the other pieces of the technologies covered in the remaining parts of the course.

The JavaScript segment is pretty much left untouched given the importance of the technology and the fact that students typically do not have this technology as a potential background entering the course. JavaScript is the still the most prevalently used technology for client-side scripting and thus its place in the course is in tact.

The Java segment is trimmed a little bit as compared to previous offerings. This is not because the technology is viewed as one that is less important now. In fact, it is the complete opposite! Students are actually getting more doses of the language in other courses (e.g., Object-Oriented Systems Development) and thus its treatment in the course is reduced. Note that it is still a significant portion of the course given that some students may not have seen Java before entering the course. For the ones who have had Java background before, this segment gives a slightly different look at the technology (more web emphasis) and a different set of assignments.

One of the most significant changes to the course (when comparing the present offering to the past) is that the CGI piece is almost completely revamped. (The only mention of CGI is a note on its historical importance in web development evolution.) Instead of using CGI, Java servlet technology is used. Servlets replaces the need to cover CGI since it behaves much like a CGI program but it is implemented in Java (thus no new language/environment needs to be introduced) and it is more efficient. Unlike a CGI program, once a servlet is loaded it need not be reloaded every time a request is made to the program. (A new process would need to be spawned each time if the request was for a CGI program.) Lastly, it has all the benefits that are inherent in the Java technology such as architecture independent, security, etc.

For separating presentation logic from business logic, JSP (Java Server Pages) is briefly discussed in the course. Because of the brevity in coverage, students are not required to integrate this technology into their projects (see Section 3 below).

A new component has also been added to the current offering of the course. It is the all ubiquitous XML (eXtensible Markup Language) technology! This technology has mushroomed so quickly in the industry that a web development course without XML just does not seem right. Here, students are exposed to the benefits of XML as a standard for data exchange and its syntax and semantics.

Lastly, the web databases component has also been reworked in that the ADO technology has been completely substituted with JDBC (Java Database Connectivity). One of the main reasons for this switch is that it is felt that the transition to this technology (with Java as the preceding course component) can be done much more smoothly. The potential drawback of this move is that the course is marching more and more towards the "Java camp."

2.3 (Near) Future
In the near future (one cannot predict any further than that!), it is anticipated that the HTML$^+$ component coverage will continue to shrink. This is due to the fact that a lower level core course has successfully piloted the introduction of a mini HTML component (mostly the markup language and not the web architecture itself) for the submission of students' work. Thus, it is envisioned that only the protocols of the web will be discussed in the new component, more appropriately entitled *Web Architecture Introduction*.

The JavaScript component is likely to remain the same whereas the Java component will receive an overhaul. For the former, it is the same reason mentioned above – JavaScript is still an important web technology and it is not likely to be a language that students will have as a prerequisite before coming into the course. For the latter, the reason is because the two semester programming sequence in the core, which is shared by all degree programs, is now teaching the OO paradigm using C++. Thus, given the syntactic similarity between C++ and Java, much of the Java segment that deal with the syntax of the language can be dramatically trimmed. The OO Concepts component can also be eliminated for the same reason. It should be noted that the semantics of the language still need to be included as C++ and Java do differ significantly in terms of language semantics.

For server-side programming, note that the servlet component will be changed somewhat in that JSP technology is to be fully integrated into the component. This gives a more comprehensive

coverage of technologies that are commonly used in developing N-tiered web applications. The web databases (via JDBC) component is likely to remain the same but the XML component is envisioned to include a number of its related technologies. They include XSL (eXtensible Style Language) and XSLT (XSL Transformation), two important web technologies that allow XML to be used in B2B (Business-to-Business) applications and in environments where different devices can be used to browse the web.

The latest addition to the core materials in the next incarnation of the course is the topic of Web Services (Benfield '01, Gisolfi '01, Kiely '01, McCright '01). This seems to be the hottest buzzword in the industry right now and given the strong support for this technology from major players such as Microsoft, IBM, HP, and Sun Microsystems, it is projected to be a strong technological topic that students from this course ought to be exposed to.

In a nutshell, Web Services can simply be thought of as self-describing services that are URL-addressable. The technology relies on SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language), and UDDI (Universal Description, Discovery, and Integration) as the underlying technologies for involved parties to communicate and produce/consume a web service, as shown in Figure 3. Here, a scenario that shows a brokerage house registering its stock quote service with a registry and a financial software finding and consuming the service is depicted. Behind the scene, SOAP uses HTTP as the protocol to transmit its message, which is in XML format. Given that HTTP is a firewall friendly protocol and that XML is becoming more and more popular as a standard for transmitting data, it is not surprising that Web Services has received so much attention in the industry, including a brand new Web Services Journal for the coverage of the topic (Web Services Journal '01). It is also the center piece of the .NET campaign of Microsoft Corporation (Gates '01).
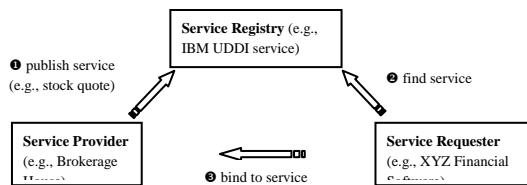

Figure 3: Life Cycle of a Web Service Execution (Registry, Lookup, and Consumption)

2.4 Tools Used

For all the offerings of the course, students are given the freedom to use any tools they wish to implement the various features identified in the assignments. Thus, one can for example use a plain editor to code the HTML tags or use Microsoft FrontPage or Macromedia Dreamweaver for the WYSIWYG effect. They are simply reminded to know what necessary tags (i.e., what are under the hood) for exam purposes. The same goes for the JavaScript component.

For the Java and JDBC components, Sun's Java SDK (Software Development Kit) has been used. However, students are encouraged to use one of the IDEs installed in the computer labs, namely IBM VisualAge for Java, Sun Forte for Java, and more recently Borland JBuilder. Virtually no lecture time is spent on the IDEs.

For servlets, Sun's JSDK (Java Servlet Development Kit) has been used as the primary development environment. Brief discussion was also given to Apache's Tomcat Server. More recently, Allaire's JRun application server is the chosen hosting environment for doing servlet and JSP development.

As for XML, plain editor and Microsoft's XML Notepad were used for coding XML documents. More recently, Altova's XMLSpy, an XML IDE, is integrated into the course so that development in XML and its related technologies can be more easily accomplished. The students will also be steered toward other IDE (e.g., JBuilder) for other kinds of development in the course (e.g., JSP).

Lastly, for Web Services component, many technologies and tools that support its development are currently being experimented (e.g., in (Baby '02), where IBM's WSDE (Web Services Development Environment) (IBM '01) and Microsoft's SOAP Toolkit (Microsoft '01) are used for developing web services). Sun Microsystems also has a Web Service development toolkit currently on early access release (Sun '02). One or more of the above will be selected for use in future offering of the course.

It should be noted that all of the above are tools can be acquired for free to educational institutions. SDK, JSDK, JRun (Developer edition), JBuilder 5 Personal, XML Notepad, Tomcat, Forte for Java (Community edition), and the Web Services toolkits are free to the general public. VisualAge for Java and XMLSpy are free to qualified educational institutions.

3. HOMEWORK, ASSIGNMENTS, AND COURSE PROJECT

A student's grade in the course is evaluated based on

his/her performance in the areas of tests (2-3 of them, worth approximately 45% of the course), "tiny" homework (worth 10%), "small" assignments (worth 25%), and a "large" project (worth 20%). These categories (the latter three) correspond to this author's philosophy about teaching reinforcement. They follow a pedagogical pattern (Lim 96) that illustrates how the use of reinforcements of various "sizes" can be an effective teaching approach, especially for the topic of information technology. It stresses the importance of a 3-stage reinforcement of the lectures presented in class. The three stages permit an instructor to monitor student progress in a topic-by-topic basis, to test if the students can combine the topics and apply them in a larger setting, and to solve a "real-world" problem using all the concepts discussed (and thus seeing the "big picture"), respectively.

For the "tiny" portion, the homework exercises are designed to simply reinforce the basic concepts covered in class. The intent of these exercises is for the students to practice straight retrieval of knowledge discussed in class. These types of exercises are typically due the next class period as not a whole of thoughts need to be involved. Examples of homework in this area include the implementation of an inheritance relationship and overriding in the Java programming language, the validation of an HTML form using JavaScript, etc.

As for "Programming in the Small," assignments in this category require the students to be able to assimilate various concepts discussed in class and put together a plan to tackle the problem at hand. They are typically given a week or two to turn in assignments of this kind. Examples include building a web quiz using JavaScript and a simple shopping cart web site using servlet/JSP technologies.

For the course project, students work in groups of typical 4 persons. The intent of this project is to allow the students to see "Programming in the Large" so that they are more prepared to join a project in a "real world" situation. This also represents an opportunity for the students to gain group experience as part of their undergraduate studies.

In this project, students are to also get a culminating experience on the web development technologies studied in the course. Thus, part of the requirements of the project is to incorporate and integrate the vast majority of the technologies studied in the course. While the students are not required to follow a specific architecture for their projects, they are steered toward one that resembles the following figure (Figure 4). This is very similar to the architecture proposed by McFall (McFall 98). Any major deviation from this

architecture requires permission from the instructor.

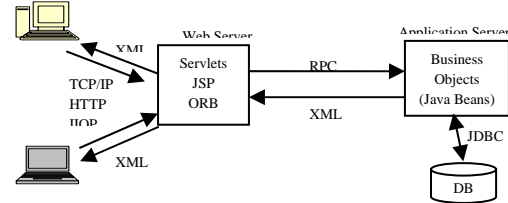Give the pace in which new (and relevant) web technologies are introduced, graduate students taking

Figure 4: Sample 3-Tier Web Application Architecture

this class are also required, as part of their assignments, to work as a group to study one of the chosen, new technologies. A class presentation is required for each of the groups. This ensures that students get exposed to technologies that this course does not have time for. It also allows the students to practice independent research and presentation skills. Possible topics include "M (Mobile) Commerce," "Internet Security," and "Internet Appliances."

4. SAMPLE COURSE OUTLINE

The following describes a sample course outline for a Web Development Technologies course that this author will use in the next offering of the course. It makes the assumptions that the students coming into this course will already have the prerequisites of (a) 2 semester of programming with objects, (b) a basic system analysis and design course, and (c) basic HTML knowledge.

5. OBSERVATIONS

In addition to the observations made regarding topical coverage in Section 2, the following observations are made from the perspectives of the students and instructor involved, as well as from curriculum design viewpoint.

First, from the perspective of the instructor, this course has been a joy to teach over the last few years. It should, however, be noted that the course consumes tremendous amount of preparation time. As many know, computer related courses, in general, require of enormous amount of time and efforts to prepare as the field changes so rapidly. This course sets a new record for that in this instructor's opinion.

Also, in terms of textbook, unlike traditional IS/CS

courses where an instructor can typically rely on a good textbook to setup the foundation of the topics to

Topics	Length	Brief Descriptions
HTML[+]	1	Introduction to the Web, Browsers, HTML, Web development architectures
Java Script	2	JavaScript vs. Java, Client-side vs. server-side processing, JavaScript object model, JavaScript applications (e.g., HTML form validation, dynamic web interactions)
Java	2	Java syntax and semantics, OOP using Java, Exceptions, Applets, Event handling, AWT and/or Swing
Servlets & JSP	2	Servlets vs. CGI, HttpServlet, doGet and doPost methods, Sessions, Separation of presentation from business logic, Scriptlets, JavaBeans, Shopping cart applications
XML	2	HTML vs. XML, well-formedness, validity, XML applications (e.g., B2B), XSL, XSLT, XML Schema
Web DB (JDBC)	2	2-tier vs. 3-tier architectures, Types of JDBC protocols, DriverManager, SQL Connection, Types of Statements, ResultSet, OO-relational mapping, Serializability
Web Services	2	SOAP, WSDL, UDDI
Others1	2	Two tests, Introduction and discussion of assignments, project, presentations, and tests
Others2	1	Catch up / Miscellaneous (Usability Design, User Interface, Human Factors, etc.)

be covered, it was impossible to find a textbook that covers the aforementioned topics when this course was first offered. When a textbook that was close in covering the topics was identified, one can easily see that significant portions of the information presented in print are already out-of-date. Also, since this course covers a number of major topics, it was hard to single out one book that does the job. A book that deals with all of the topics tends to discuss them in breadth (oftentimes superficially) and not in depth. Overall, it has been a challenging task to gather together all the relevant resources (from the web, journals, trade magazines, hot-off-the-press books, etc.) to supplement a textbook. Two recently published books

may help alleviate this challenge – they are *Core Web Programming* (Hall '02) and *Internet & WWW: How to Program* (Deitel '02).

From the students' point of view, this course provides a means to jump on the web bandwagon. It represents a course that many of them can easily relate to as they personally are the users of many Web commercial systems, the kinds that they learn to develop in this course. They often are anxious to get their assignments going because they can get graphical, animated, multimedia-based, database-intensive, network-based programs running quickly through reusable components. They generally have very positive experience with the course.

From the curriculum design point of view, both the IS subsequences in this author's department share a number of common courses currently (a total of 30 major hours). Of these common courses, two that have been closely scrutinized are COBOL and JCL/VSAM. Many of the faculty feel that they are out of place in the Web Application Development sub-sequence and should be replaced with other, more web related courses (e.g., a more advanced web course, a web security course). On the other hand, many also feel that since the subsequence is still under the IS degree umbrella, those two courses are the "sacred cows" and should not be removed. Another concern about removing the two courses is that many students will flock to the new subsequence because of the perceived benefits and personal interests. If that theory holds out, faculty resources (to teach the courses in the new subsequence) will be a huge problem. The debate goes on for this particular issue.

6. SUMMARY, CONCLUSION, AND FUTURE WORK

Information technology is changing very rapidly. To prepare graduates from IS/CS curricula so that they can be productive in the 21st century, the curricula must be upgraded to incorporate the latest advances in information technology. The web, with its related technologies, is a gigantic topic of study about information technology that no educators should ignore.

This paper describes a web development technologies course that represents this author's attempt to keep students abreast of the state-of-the-art technologies in the field. It is a course that this author has taught over the last couple of years. The paper highlights the evolution of the course, from the original version about 4 years ago, through the current incarnation, to the (near) future version.

If there were another few weeks that could be added to the course, then many other equally important topics could be added. They include topics such as Microsoft's .NET architecture, Enterprise Java Beans (EJBs), more XML-related technologies (the sample course outline above allows for only a cursory coverage of surrounding XML technologies), and wireless technologies (e.g., WAP (Wireless Application Protocol), WML (Wireless Markup Language)).

This author has interviewed many in the industry, including members of BIAC (Business Industry Advisory Council) associated with his department, and the vast majority of them have indicated that the above technologies are important technologies that students should know in a web development sequence.

Perhaps another 16 weeks worth of materials (i.e. a second web development technologies course) ought to be introduced? That incidentally is the finding of a recent discussion during a departmental retreat and this author is to pilot this advanced web course in the upcoming semester.

7. REFERENCES

Baby, Roshni., *Building an LDAP Web Service Using C# (.NET) and Java: A Detailed Comparison*, M.S. Project, Applied Computer Science Department, Illinois State University, Spring (expected), 2002.

Benfield , S., "Web Services: XML's Killer App," *Java Developers' Journal*, Vol. 6, No. 4., 2001.

CNNFn, "Internet Access Soars," http://cnnfn.cnn.com/2001/08/09/technology/internet_fcc/, Retrieved August 9, 2001.

Deitel, Harvey M., Paul J. Deitel, and T. R. Nieto, *Internet & WWW: How to Program*, 2nd Edition, Prentice-Hall, 2002.

Gates, W., *Microsoft .NET Today*, an open letter to the Developers & IT Professionals, June 14, 2001.

Gisolfi, D., "*Web Services Architect (Part 1-4): An introduction to Dynamic e-Business*," http://www-106.ibm.com/developerworks/webservices/library/ws-arc1/?dwzone=webservices, 2001.

Hall, Marty. and Larry Brown, *Core Web Programming*, 2nd Edition, Prentice-Hall, 2002.

Kiely, D., "WSDL for Defining Web Services," Cover Story, *XML Magazine*, Vol. 2, No. 4, August/September, 2001.

IBM Web Services ToolKit, IBM Corporation, http://alphaworks.ibm.com/tech/webservicestoolkit.

Lim, B. L., "Effective Object Pedagogy Patterns: The Programming in the Tiny, Small, and Large (TSL) Pattern," *Workshop on Object Pedagogy Patterns, International Conference on Object-Oriented Programming Systems, Languages, and Applications*, October, 1996.

Lim, B. L., "Teaching Web Development Technologies in CS/IS Curricula", *ACM SIGCSE*, Vol. 31, No. 1, March, 1998.

McCright, A., "Writing Your First Web Service: A Tutorial," Web Services Journal, June, 2001.

McFall, C., "An Object Infrastructure for Internet Middleware," IEEE Internet Computing, Vol. 2, No. 2, March-April, 1998.

Microsoft Web Services SOAP ToolKit Microsoft Corporation, http://msdn.microsoft.com/downloads/, 2001.

Sun Web Services Development Toolkit, http://java.sun.com/webservices/, 2002.

Walker and Browne, "Teaching Web Development with Limited Resources," *ACM SIGCSE Bulletin,* 31:1, March, 1999.

Web Services Journal, Sys-Con Publications Inc., Montvale, New Jersey.

Wired Magazine, April, 1996.

AUTHOR BIOGRAPHY

Dr. Billy Lim is a tenured, associate professor in the Applied Computer Science department of Illinois State University. He received his B.S. degree in Computer Science from the University of Toronto, Canada and M.S. and Ph.D degrees in Computer Science from the University of Louisiana at Lafayette, Louisiana. His research and teaching interests are in the areas of Web development technologies, object-oriented (OO) programming, and new generation database management systems. He has published numerous journal articles in these areas and also presented papers in many international, national, and regional conferences.

Information Systems & Computing
Academic Professionals

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.