THE CHALLENGES OF TEACHING WEB PROGRAMMING
Literature Review and Proposed Guidelines

Stelios Xinogalos and Theodore H. Kaskalis
Technology Management Department, University of Macedonia, Loggou-Tourpali Area, 59200 Naoussa, Greece

Keywords: Web Programming Model, Information and Communication Technologies Education, Teaching Practices.

Abstract: The main concern of this paper is the exploration of current challenges, teaching techniques, course design methods and topics covered when attempting to teach a web programming course in technology-oriented higher education departments. The authors attempted a literature review of the subject in order to identify, compare and analyse the existing experience upon which one can establish solid guidelines towards a manageable, efficient and comprehensible course model. In the paper, the presentation of the value of teaching web programming is followed by an extensive listing of the underlying challenges. Consequently, the various teaching approaches are presented comparatively and comments are made as regards the topics covered and the tools used. The study and analysis of this gathered experience naturally leads to certain outcomes. An extensive list of questions is summed in a list view and this list aims to help educators of the field to prepare the respective structure, content, methodology and tools of a web programming course that will serve their needs in a productive way. The authors' intention is to build upon this knowledge towards a web-based comprehensible web programming environment that will aid the process of teaching this challenging subject.

1 INTRODUCTION

It is more or less a given fact: web programming is almost mandatory in a Computer Science or Information Technology curriculum. The web as a platform offers significant job opportunities and has gathered around it a great deal of attention. Statements such as "the web has won; it has become the dominant programming model of our time" (Gundotra, 2009 as cited in Hollingsworth and Powel, 2010) and "no major desktop application has been released since 2004" (Gundotra and Picai, 2010 as cited in Hollingsworth and Powel, 2011) are becoming respected by the majority of the IT industry.

Isolating job opportunities, the numbers appear quite promising. Cloud computing is expected to add 2.4 million jobs in Europe's biggest economies by 2015, according to a December 2010 report by the London-based Centre for Economics and Business Research (as cited in Patel, 2011). 10.000 jobs are expected to appear by Facebook, Google, Twitter and Zynga (Patel, 2011) and the title of America's Best Job was given to 'Software Engineer' for 2011 (Strieber, 2011). Programming for the web is a rapidly growing demand in job offers and we have to respect that. Moreover, the web as a platform reaches more and more embedded devices which pave the way towards facing services as cloud based (or browser based) only. The emerging standards and technologies strongly suggest a shift towards internet-only based services.

It therefore comes as no surprise that computer technology related departments increasingly orient their course of studies towards web programming techniques. However, is this a simple and straightforward task? The answer cannot come that easily. One has to identify the topics that need to be addressed, the maturity of current web technologies, the difficulties in learning this particular model, the trends that seem to be formulated very fast, the teaching methodologies that have emerged (or need to be emerged) for this subject and the environment that will bring everything together in a comprehensible and manageable way. Through this paper we attempt to analyze the many aspects that need to be considered in a web programming course and we wish to state certain conclusions and ideas that will bring forward this ever-changing subject.

We attempted an extensive literature review in the matter and the outcomes seem quite intriguing. Web programming is a real challenge and should be

Xinogalos S. and H. Kaskalis T..
THE CHALLENGES OF TEACHING WEB PROGRAMMING - Literature Review and Proposed Guidelines.
DOI: 10.5220/0003960902070212
In *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (WEBIST-2012), pages 207-212
ISBN: 978-989-8565-08-2

treated as one. In the following we present our findings and propose our ideas in this order: first we examine the main challenges that seem to exist in a web programming course. Then we present various teaching approaches and we turn our attention to the design of a respective course. Following, we gather a series of questions that need to be addressed in order to proceed to certain proposals about the structure of a potential web programming course. Finally, we summarize our findings and conclude with our future research in this subject.

2 CHALLENGES OF TEACHING WEB PROGRAMMING

The development of a web application demands the combination of various technologies. Moreover, the available choices for both client-side and server-side web development technologies are numerous (Chung and McLane, 2002), some of them complementary and other competing (Treu, 2002). *Selecting the topics* and the underlying web technologies covered in a web programming course, as well as *providing the appropriate depth* for each topic is the first challenge faced by an instructor (Liu and Phelps, 2011).

Usually, a wide range of topics is selected, which makes it *difficult to find a single textbook that is appropriate for the course* (Hollingsworth and Powell, 2011; Yue and Ding, 2004; Noonan, 2007; Stepp, Miller and Kirst, 2009; Treu, 2002; Wang, 2006). Some textbooks provide depth but lack breadth, while others provide breadth and lack depth (Noonan, 2007). Moreover, some good textbooks are targeted to more advanced courses (Stepp et al., 2009) or even to professionals (Yue and Ding, 2004; Lee, 2003). Since, it is not possible to require several textbooks – due to the economical burden it brings on students – several instructors rely on resources that are available through the web. However, this is problematic too. Although, there is an abundance of web tutorials, the vast majority are sloppy, outdated/obsolete, and even wrong as templates to rely on (Yue and Ding, 2004; Stepp et al., 2009). Unfortunately, it is not easy for students with little experience on web programming to evaluate the timeliness, correctness and relevance of resources available on the web (Yue and Ding, 2004). On the other hand, determining the most appropriate web tutorials is a time-consuming task for professors, the most difficult one according to Treu (2002). In most cases, students are involved in one or another way in evaluating web resources. For example, students may be assigned to recommend online tutorials or articles that are approved by the instructors prior to class use (Yue and Ding, 2004). Alternatively, students may be provided with a list of online resources and asked to submit new links, as well as reviews regarding the help provided by specific resources, so as to formulate a list of resources that are the best for students to utilize (Treu, 2002). These web tutorials are usually utilized in combination with material prepared by instructors. As a matter of fact, in most cases, instructors decide to use three different resources for supporting students from different perspectives (Walker and Browne, 1999; Yue and Ding, 2004):

- A *main textbook* covering some of the topics considered to be in the core of the course
- *Lecture notes* and laboratory *examples* developed by instructors
- Freely available *web resources* selected by instructors, or proposed by students and approved by instructors.

The way that each of these resources is utilized is different in each case. For example, Treu (2002) used the web resources as the main resource for the course, while Lee (2003), Walker and Browne (1999) used a textbook along with tutorials and references found on the Internet. Yue and Ding (2004) consider that the best teaching materials are written by the instructors themselves, since they are developed for fulfilling the goals of the course. So, Yue and Ding (2004) used their own materials as the key resource and the main textbook was used as a secondary resource. When it comes to textbooks the most referenced ones are the books by Hall and Brown (2003) and Hall, Brown and Chaikin (2007) that refer to servlets and JSP (Hollingsworth and Powell, 2010; Lee, 2003; Yue and Ding, 2004). The relevant site maintained by Marty Hall (http://www.coreservlets.com) is also one of the most referenced ones. Useful resources for web development technologies are available from W3C.

After selecting the course topics and preparing the appropriate educational material the next step is *configuring and managing a web programming environment*, which is one of the most difficult aspects of web programming (Amon, 2003). Amon (2003) mentions that student actions, while working on servlets and JSP, often resulted in crashing the Tomcat server used in the course. Lee (2003) gives emphasis on the inadequate documentation on web servers. In some cases, dealing with a problem for both Apache and Tomcat servers, requires reading the source code. Hollingsworth and Powell (2010) describe a novel approach for teaching a web

programming course and dealing with the problems mentioned above. Specifically, they propose using Google's App Engine (http://code.google.com/appengine) for developing and deploying Java servlets and JSP. Using the Google Cloud means using Google's infrastructure – hardware, software, storage – for deploying web applications. This service is free and alleviates the problems related to the acquisition, configuration and administration of a course server, or the configuration of a server locally at students' systems. Furthermore, the Google's Plugin for Eclipse (http://code.google.com/appengine/docs/java/tools/eclipse.html) that is used as a programming environment lets students create, test and upload App Engine applications from within Eclipse.

Even when the aforementioned steps have been taken and a plan for a course has been established, adjustments might be necessary. These adjustments might be a side effect of the instructor's incorrect or inadequate understanding of the various technologies, as well as the fact that for the same purpose a variety of unfamiliar topics/technologies might be utilized (Lee, 2003). However, the most common reason for adjustments or, worst, changes to a course is the fact that *web technologies change rapidly*. New technologies and standards emerge and adopting them in an existing course is not an easy task. The instructor has to prepare new educational material, familiarize with new software tools and make critical decisions regarding the appropriate breadth and depth devoted to each technology. In addition, problems regarding backward incompatibility and lack of support from browsers have to be taken into account. In order to cope with the problem of keeping up with the large number of continuously changing web technologies and the aforementioned consequences, Treu (2002) has proposed adopting a seminar format for the course. Specifically, students have to learn and teach topics to their classmates. As Treu (2002, p. 201) states "students' collective expertise on web programming is probable to supersede the instructors' expertise", since "the strong appeal of Web programming makes it a subject for many enthusiastic hobbyists outside academia". Other researchers, such as Klassner (2000), have proposed focusing the course on concepts, which means that keeping up with the latest technologies, is not so important. It seems that focusing a course on established web technologies, that is technologies that are widely supported and frequently used, is a better choice (Liu and Phelps, 2011).

Focusing on widely supported web technologies is considered even more important, taking into account the *inconsistency between web browsers*. The inadequate integration of current web technologies and inconsistent implementation of standards in web browsers (Liu and Phelps, 2011), combined with the fact that most of them produce no output for most errors, adds complexity to the overall design of a web programming course. In order to make the complexity more manageable, several instructors choose to use a single browser. For example, Stepp et al. (2009) have chosen to use Firefox as the target browser for their elective web programming course. This decision has the advantage that students can rely on published web standards when coding, leaving aside the quirkiness of other browsers. However, in order to support students that are interested in running their applications in Internet Explorer too, Stepp et al. (2009) provide links to IE-related bugs and fixes.

No matter if the instructor decides to use a single browser or not, errors occur and *debugging programs for the web is too challenging*. Debugging is one of the most difficult and sometimes frustrating tasks for novice programmers. Error messages often use codes, do not refer the actual line of the error and use terminology that is not comprehended by novice programmers. In web programming, things are even worse, since: (1) *students have to learn many programming language-paradigms and technologies in a short time* and (2) many of the most common student *mistakes produce no output in the browser* (Stepp et al., 2009), while specialized tools for debugging often do not explain errors adequately (Liu and Phelps, 2011). Furthermore, instructors' experience on utilizing debugging tools is not consistent. For example, Stepp et al. (2009) consider Firebug to be an excellent tool, while Liu and Phelps (2011) state that errors are not explained adequately (the element that caused an error is not reported). However, the use of such tools is considered necessary from some instructors. Stepp et al. (2009) consider debugging as the most prominent difficult aspect in a web programming course and have reported on using various debugging tools, such as Firebug, W3C XHTML and CSS validators, as well as JsLint Javascript syntax checker. Moreover, Hickey (2004) mentioned working on building better debuggers for Scheme servlets and applets in order to tackle with the unnecessary frustration of some of his students. Lantis (2008), Hu and Hu (2004) have also reported on utilizing web-based teaching tools for supporting students in developing programs for the web.

3 TEACHING APPROACHES

Teaching programming courses in general is heavily based on programming assignments and projects. This is usually the case for web programming courses too. However, the type of the assignments/projects that can be used and the underlying merits, differ significantly in the following sense (Yue and Ding, 2004):

- A collection of *unrelated assignments* is used, in order to cover more technologies and concepts.
- A *tightly coupled project* comprising of a series of related assignments is used, in order for students to gain experience in building a complete web application.
- A *loosely coupled project* comprising of a series of unrelated assignments is used, in order to achieve a broad coverage of technologies in combination with building a somewhat complete web application.

Although gaining experience in building a complete web application is desirable, it is not easy to accomplish. Students have to integrate separate technologies, which is far more difficult than just comprehending them. Deciding what type of assignments and projects will be utilized in a web programming course, as well as achieving a good balance between foundations and practice (Finkel and Cruz, 1999), is quite challenging, while the literature shows that all the aforementioned approaches are applied:

Yue and Ding (2004) reported on their experience with a *tightly coupled project* in the context of their "Internet Application Development" course at Houston University. The authors concluded that there is not a best approach and they would apply a mixed approach on the next offering of the course.

Noonan (2007) describes his positive experience on a senior-level "Web Programming" course, which is centered on a *single, simplified version of a real web site project divided into six assignments*. Overall, the course emphasizes depth over breadth and principles over technology, choices that were positively evaluated by students.

Liu and Phelps (2011) use a mixed approach for their web programming course, which is offered as an elective to junior and senior CS students at Georgia College and State University. Specifically, the course is based on in-class labs, *web page assignments and a real-world web project incorporating the three common tiers* (presentation, logic and storage tier) of designing web applications and the corresponding technologies.

Wang (2006) has designed a web programming course for junior and senior CS students at the University of Texas Pan American, which is highly demanding. Specifically, students are required to take two tests and one final exam, and also to finish *eight assignments and one final project*. In order to support both instruction and students in their final project, the instructor has implemented a *sample web application* (an online bookstore web site) that is used for demonstrating all the fundamental operations of such an application. The majority of students viewed the assignments and the final project positively and specified that they "provided a good practice opportunity for web programming" (Wang, 2006, p. 220).

Gousie (2006) designed an inter-departmental "Web Programming, Graphics and Design" course for non CS majors at Wheaton College. The first offering of the course included *five projects*, with the last one being assigned to groups of up to three students. What was different with the aforementioned courses is that a portion of class time was devoted to supporting students on their projects that were not developed from scratch – *students added or modified code to an existing web page*.

Amon (2003), in accordance with Gousie (2006), believes that web programming is taught more effectively *using an existing web application as the basis for presenting web programming concepts and devising assignments/projects* for students to solve. Several projects of varying difficulty can be assigned, while students clearly prefer extending or modifying existing code according to Amon (2003).

All the aforementioned courses lie heavily on assignments and loosely or tightly projects for familiarizing students with web programming languages and technologies. Treu (2002) also uses projects, but in a novel way. He has proposed using a *graduate seminar, demand-driven and simulation-based format* for teaching the "Web-Based Application Design" course developed and taught at Furman University. The course has a form of a graduate seminar, which is based on a series of *real-world case studies* and *teamwork simulation* of real-world development processes. Since students act as employees working on a web development project, the learning is demand-driven and consequently the exact content of the course cannot be completely decided from the beginning of the course. During the course, three case studies take place and students rotate, so as to work on each one of the following teams: Design, Server-side and Database Team.

Each team gives a lecture on the technology used.

4 ISSUES TO BE ADDRESSED

The aforementioned analysis can lead us to a series of topics or questions that need to be addressed when preparing a web programming course:

- What is the instructors' background and expertise?
- Are there any relevant courses in the curriculum?
- What aspects of web programming will be covered? (client-side, server-side, databases)
- What is considered more important? Covering a limited number of topics/technologies in depth or covering the fundamentals of a breadth of topics/technologies?
- What is the mixture of materials that will form the reading prerequisites?
- Will emphasis be given on web programming concepts or technology?
- Will the course include the latest technologies or widely supported, stable and mature technologies?
- What web development tools will be used?
- Will there be a specially prepared environment or just common tools?
- What debugging tools/approaches will be utilized?
- What is the available infrastructure (existing platforms and laboratories)?
- Are there any assistants for the course?
- Based on the selection of topics and technologies, what servers will be needed? (software level)
- Is there any server (hardware level) that can be devoted to the course or students have to configure a server locally on their computers?
- Will a single target browser be used, or the web applications developed must run correctly in any browser?
- What are the anticipated students' difficulties based on their background?
- How important is the hands-on experience in creating real-world web applications?
- Which code percentage of projects and/or assignments will be pre-coded and filled by students?

Since we both teach in a Technology Management Department and are involved in computer programming courses, we will attempt to address the above issues in accordance to our specific profile. As a matter of fact, we intend to re-shape the department's curriculum and incorporate a well-founded web programming oriented course of studies.

5 CONCLUSIONS AND FUTURE WORK

Our department currently includes three distinct compulsory courses and one elective that deal with web programming technologies. The compulsory courses appear all together in the spring semester of the second year of studies and address (1) data bases, (2) server-side technologies and (3) client-side web page creation and programming, respectively. It must be noted that, up until this 4th semester, students have been exposed to C and Java programming in 2 respective previous courses. The 8th semester elective course extends students' knowledge in Java programming, with its web programming aspect including applets and servlets. Moreover, it must be noted that our degree title and curriculum orientation aims towards managers of Information Technology (IT) and not core computers scientists or engineers. As a result, we care more about the proper understanding of fundamental notions and less about deep expertise in one technology.

With these in mind and while we strive towards a web programming course (or maybe mixture of courses) we address the above issues for our case. Including ourselves, we form a group of 4 instructors with experience in HTML, XHTML, CSS, JavaScript, PHP, SQL and the Document Object Model along with Ajax techniques. Relevant courses are considered those about programming on C and Java and our main aim is to cover all three tiers of the web multitier architecture model in a clear and transparent way. We want our students to fully understand "what runs where".

We choose depth over breadth about the aforementioned technologies, but we will include discussions about similar approaches. We intend to prepare our own materials along with a special web-based environment that will incorporate novel methodologies borrowed from our long experience in teaching programming classes. In fact, we intend to conduct further research in applying the microworld notion into web programming.

Our emphasis will equally fall between concepts and technology and we choose the stable and mature product adoption. The development tools will be inside a specially prepared web-based environment with "cut-off" versions of popular tools, in order to

raise the burden of teaching many languages in full depth and achieve easier debugging. Our environment will reside in one of our department servers and will be accessible remotely, while all the implemented code will be available as a download.

We select to fully respect the existing open standards, as regards browser compatibility, and we expect significant student difficulties in the matters of problem solving abilities. This is why we intend to "nurse" them in a controlled environment and rely heavily on semi-complete code fragments. On the other hand, we will include a full-scale web application as a final project, and we intend to tempt our students with 2 nation-wide contests about new business ideas: the most prominent ideas will be formulated to produce the project of the class.

As a conclusion, with this paper we wish to communicate our findings and remarks and state our intention in developing a new environment that will aim in teaching web programming through new methods. The expected discussion from the community will help us formulate our final decisions and open a research area in addressing web programming models with new tools such as those found in microworld environments.

REFERENCES

Amon, T. (2003). Bicycle club mileage log: a servlet application for teaching web programming. *Journal of Computing Sciences in Colleges*, 19(1), 219-226.

Chung, W. S. and McLane, D. (2002). Developing and enhancing a client/server programming for Internet applications course. *Journal of Computing Sciences in Colleges*, 18(2), 79-91.

Finkel, D. and Cruz, I. F. (1999). Webware: A course about the web. *Proceedings of the 4th annual SIGCSE/ SIGCUE ITiCSE conference on Innovation and technology in computer science education*, 107-110.

Gousie, M. B. (2006). A robust web programming and graphics course for non-majors. *ACM SIGCSE Bulletin*, 38(1), 72-76.

Gundotra, V. (2009). *Google I/O 2009 - Keynote Day 1*. Retrieved January 20, 2012, from http://www.youtube.com/watch?v=S5aJAaGZIvk

Gundotra, V. and Picai, D. (2010). *Google I/O 2010 - Keynote Day 1*. Retrieved January 20, 2012, from http://www.youtube.com/watch?v=a46hJYtsP-8

Hall, M. and Brown, L. (2003). *Core Servlets and Javaserver Pages: Advanced Technologies* (2nd ed., Vol. 1). Upper Saddle River, NJ: Prentice Hall.

Hall, M., Brown, L. and Chaikin, Y. (2007). *Core Servlets and Javaserver Pages: Advanced Technologies* (2nd ed., Vol. 2). Upper Saddle River, NJ: Prentice Hall.

Hickey, T. J. (2004). Scheme-based web programming as a basis for a CS0 curriculum. *ACM SIGCSE Bulletin*, 36(1), 353-357.

Hollingsworth, J. and Powell, D. J. (2010). Teaching web programming using the Google Cloud. *Proceedings of the 48th Annual Southeast Regional Conference*, doi:10.1145/1900008.1900110

Hollingsworth, J. and Powell, D. J. (2011). Requiring web-based cloud and mobile computing in a computer science undergraduate curriculum. *Proceedings of the 49th Annual Southeast Regional Conference*, 19-24. doi:10.1145/2016039.2016054

Hu, J. H. and Hu G. M. (2004). Teaching web design and web programming using an enhanced teaching environment. *Proceedings of the 2nd annual conference on Mid-south college computing*, 148-154.

Klassner, F. (2000). Can Web development courses avoid obsolescence? *ACM SIGCSE Bulletin*, 32(3), 77-80.

Lantis, M. J. (2008). Using a web editor as a development platform for teaching HTML and client-side programming in the internet 101 course: nifty tools and assignments. *Journal of Computing Sciences in Colleges*, 24(1), 97.

Lee, A. H. (2003). A manageable web software architecture: searching for simplicity. *ACM SIGCSE Bulletin*, 35(1), 229-233.

Liu, Y. and Phelps, G. (2011). Challenges and professional tools used when teaching web programming. *Journal of Computing Sciences in Colleges*, 26(5), 116-121.

Noonan, R. E. (2007). A course in web programming. *Journal of Computing Sciences in Colleges*, 22(3), 23-28.

Patel, P. (2011). Where the jobs are: software engineering. *IEEE Spectrum*, 48(9), 28.

Phillips, J., Tan, J., Phillips, M. and Andre, N. (2003). Design of a two-course sequence in web programming and e-commerce. *Journal of Computing Sciences in Colleges*, 19(2), 208-217.

Stepp, M., Miller, J. and Kirst, V. (2009). A "CS 1.5" introduction to web programming. *ACM SIGCSE Bulletin*, 41(1), 121-125.

Strieber, A. (2011). *The 10 best jobs of 2011*. Retrieved January 20, 2012, from http://www.careercast.com/jobs-rated/10-best-jobs-2011

Treu, K. (2002). To teach the unteachable class: an experimental course in web-based application design. *ACM SIGCSE Bulletin*, 34(1), 201-205.

Walker, E. L. and Browne, L. (1999). Teaching Web development with limited resources. *ACM SIGCSE Bulletin*, 31(1), 12-16.

Wang, X. (2006). A practical way to teach web programming in computer science. *Journal of Computing Sciences in Colleges*, 22(1), 211-220.

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