2006-2389: INTEGRATING SCRIPTING PROGRAMMING LANGUAGE INSTRUCTION INTO IT CURRICULA

Gordon Romney, Brigham Young University

Ronald Gonzales, Brigham Young University

Integrating Scripting Programming Language Instruction into IT Curricula

Introduction

The objective of this paper is to offer IT academicians one potential open-source solution to the escalating challenge that confronts each of us regarding how to optimally teach Scripting Programming Languages [1] (Web scripting language "WSL") within severely constrained teaching curricula. The outcome, additionally, is to promote creativity within academia that will generate the correct instructional solution for a given university situation. Specifically, this paper is for those who desire 1) their students to be as well-prepared, as possible, for post-graduate employment; and 2) their program to receive positive feedback from industry for preparing quality students.

Web scripting languages (WSL) constitute a specific group of programming languages that are application-specific (browsers for the Internet), and are used to program Web pages and to control Internet usage. WSL tend to favor rapid development over efficiency of execution, and, normally, are interpretive rather than compiled languages.

WSLs addressed in this study are shown in Figure 1.

Tool	Popularity of Usage	% Usage
ASP.NET	6	0.5
Java (JSP)	1	22
Perl/Mason	3	7
PHP	2	9
Python	4	3
Ruby/Rails	5	0.5

Figure 1 Web Scripting Languages

WSLs, in Figure 1, are extracted, by the authors, from the published survey by Tiobe Software [2] from the top fifty programming languages, and their ranking by Tiobe Popularity Index and the percentage of usage is shown.

Computing Instruction Challenges

A variety of academic computing instruction challenges are addressed in this paper, namely:

1. Several parallel, competing computer instructional programs exist at a given institution of higher education, e.g. Computer Science (CS), Electrical and Computer Engineering (ECE), Information Systems (IS), and Information Technology (IT).
2. Each computer instruction program, frequently, is constrained by a total number of core instruction credit-hours permitted by its college or university.
3. Computer technology, both hardware and software, continues to grow at seemingly unmanageable rates.

Faculties are challenged to keep pace with this growth.
4. Programs, usually, are not focused on or always aware of current industry needs.
5. Four-year programs, frequently, are more constrained and cannot quickly adjust to industry-indicated trends.
6. Computing instructional programs often leave much of the learning of web-scripting languages to students' individual self-instruction.

Hypotheses for Research

Research addressed in this paper focuses on two hypotheses:

1. Show that industry is a partner desirous of providing meaningful feedback that will help higher education to better focus on industry's immediate WSL needs.
2. Substantiate that WSLs are predominately learned by self-instruction.

A Proposed Solution

A three-pronged approach is addressed and illustrated in this paper:
1. Actively involve industry in order to better understand its dynamically changing needs.
2. Maintain knowledge of WSL usage, trends and growth.
3. Constantly seek opportunities to integrate WPL usage in instruction, homework, lab and project activities.

Experience of one of the authors, who served for eight years as chair of the education committee of a state IT industry association, indicates that industry is a most willing partner in indicating its urgent skill needs. Hence, academia needs to better communicate with its industry contacts and the potential employers of its students. This helps academia better focus on market demands and not just on what it "thinks industry needs."

Given the Computing Instruction Challenges to academia, enumerated above, few programs can augment core course instruction with "current" WSL trends and demands. Elective courses are a possibility but faculty constraints make this solution difficult. Working WSL key concepts into course instruction; and lab, homework, and project assignments becomes the more viable solution. Examples will be given in this paper.

Rationale for PHP Example

A candidate WSL was sought that would best exemplify the instruction challenge that confronts higher education. Our IT program was already confronted by its IT industry contacts with a declared "immediate" need for PHP skills that, now, is over two years old.
We, likewise, were aware of the surging demand for open source solutions. A request that came from both established Fortune 500 companies as well as startup IT ventures. Were the requests that we received, however, truly indicative of market trends?

Published Articles Indicate an Explosive Growth of PHP

Open source products dominate the web server product space [3] in terms of number of domains that use a given product. Apache accounts for 72% of all web servers. And, interestingly

enough, the most popular open source product functioning under Apache is PHP, accounting for usage on 43% of all Apache sites [4]. PHP, however, is not restricted to a non-Windows platform as it has become the most popular non-Microsoft WSL even on Windows [5].

PHP usage growth has occurred, explosively, in the last six years as shown in Figure 2, a graph produced by Netcraft [6].

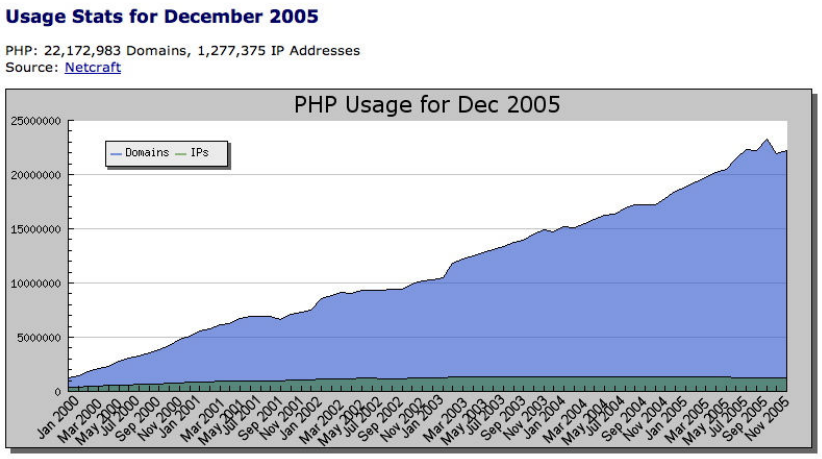

Figure 2 PHP Growth

Using a Least Squares fit to this PHP Growth curve produces a slope of 3.23 with a linear fit as shown in Figure 3.

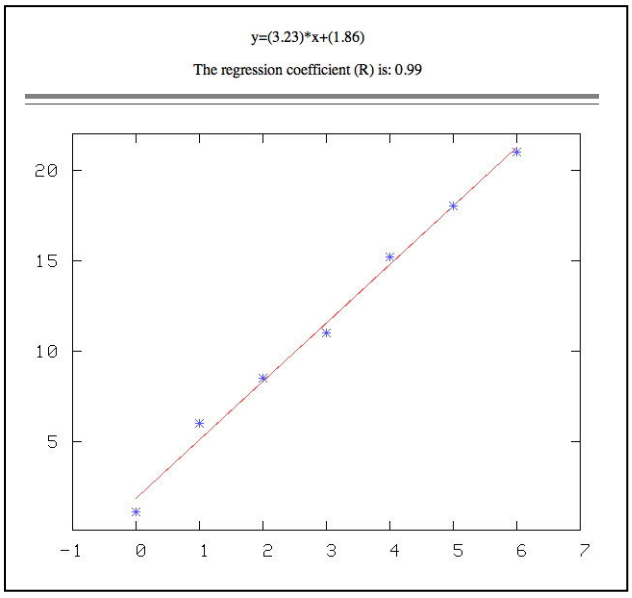

Figure 3 PHP Least Squares Fit

Major Industry Players Endorsed PHP
Yahoo! switched its enterprise level service to an open source solution using PHP in 2002. A year later it reported that the transition had been most successful [8]. This marked the beginning of the usage of PHP by enterprise-level organizations rather than the traditional smaller Web implementer using PHP. Lufthansa, IBM and Oracle have all, subsequently, endorsed the usage of PHP and the latter two have joined in efforts to "harden" and make more solid the PHP offering. Consequently, additional pressure has been applied to higher education to help meet the demands for quality WSL, and PHP, in particular, expertise.

Research Objectives for this Paper
Research focused on two hypotheses:

1. Show that industry is a partner desirous of providing meaningful feedback that will help higher education to better focus on industry's immediate WSL needs.
2. Substantiate that WSLs are predominately learned by self-instruction.

Additionally, an attempt was made to identify current and future trends in WSL instruction.

Web Programming/Scripting Languages Survey for Academia
A survey was emailed to academic contacts asking for a response to several questions that relate to WSL growth and the manner in which this growth is handled in IT instruction. The seven questions are shown below. A ranking of 1 (high) to 6 (low) was requested for the six WSLs listed in Figure 1.

1. Web tools ranked by importance in "teaching" *Curriculum*
2. Web tools employed by students to fulfill class/lab needs ranked by usage
3. Web tools ranked in order of perceived current *Industry* usage
4. Web tools ranked in order of perceived *Industry* usage in three years
5. Indicate % student learned outside of *Curriculum* for each
6. Do colleges need to improve focus of web language instruction? (Y/N)
7. Does your contact with industry indicate satisfaction with quality of your graduates' Web programming skills? (Y/N)

Question	ASP	JSP	Perl	PHP	Python	Ruby
1.Importance in "teaching" *Curriculum*	2.6	2.1	3.5	2.0	4.5	6.0
2. Used to fulfill class/lab needs	2.6	2.6	3.5	2.3	4.0	6.0
3. Current *Industry* usage	2.7	2.0	3.5	1.4	4.8	4.5
4. *Industry* usage in three years	2.0	2.1	3.6	1.9	4.4	4.2

Figure 4 Survey Response Averages Questions 1 – 4

Responses were received from seven major IT institutions and are averaged as shown in Figure 4. The smaller the average, the higher the ranking. Thus, it can be seen that the order of overall ranking yields (1) PHP, (2) JSP, (3) ASP, (4) Perl, (5) Python and (6) Ruby. Regarding both PHP and JSP it is noteworthy that academicians score, both current and future (three years hence), industry usage as being greater in importance than the ranking perceived for IT computer instruction.

Question 5 relates to the extent students learn a WSL outside of the IT curriculum.

WSL	Percentage Self-taught
ASP.NET	33
Java (JSP)	26
Perl/Mason	53
PHP	33
Python	77
Ruby/Rails	78

Question 6. Do colleges need to improve focus of web language instruction? The response was close to divided with 56% responding "Yes" to 44% responding "No."

Question 7. Does your contact with industry indicate satisfaction with quality of your graduates' Web programming skills? 67% responded "Yes" to 44% responding "No."

Industry Survey

A less comprehensive survey was sent to industry partners and only a small sample of five indicated 100% felt that Colleges Need to Improve Focus of Web Language instruction. This same sample indicated that virtually 100% of their programmers were self-taught on WSLs.

One of the respondents had given a PHP exam to one of our graduates who did not perform satisfactorily. This exam, in turn, was shared with us and did much to precipitate this paper. It definitely revealed that industry expectations were higher than we anticipated. Furthermore, the assumption that student self-instruction was sufficient we now understand to be far from satisfactory. This has led to an experiment to incorporate many of the concepts shared with us by this industry exam into our senior level security course instruction which will be explained.

Industry PHP Exam

Congruent with the attempt to establish a measure of industry entry level skill standard for PHP competency, is a 36 question test that has been used as a pre-screening evaluation for a web programming applicant by a web company. An 80% correct response would be considered as a minimum competency measure representing the necessary PHP skill level competency.

Baseline Technical Evaluation - PHP Skill Level Competency 1.0

1. What does the PHP function **print_r()** do?
2. What will this PHP statement accomplish: **echo $$var;**
3. What services does the PHP CURL library provide?
4. Write some PHP code that creates an empty array.
5. What are some drawbacks to using cookies?

6. What does the PHP **header()** function do?

7. Which PHP function sends emails?

8. What does the PHP function **unlink()** do?

9. When a file is uploaded via an html form, how can the original name of the file be retrieved in a server-side PHP script?

10. Give a text string that would be a pattern match for the following regular expression: **"[0-9]+/[A-Za-z]+/[0-9]+"**

11. Explain the PHP function **flush()**. What may keep it from successful execution?

12. How do you find the version number of the existing PHP installation?

13. How can PHP execute an external program?

14. When should a semaphore be used?

15. How can PHP sessions be made more secure?

16. When is a good time to use the PHP function **sprintf()**?

17. What are the differences between the PHP functions **echo** and **print**? Which is faster (if only marginally)?

18. What does the function PHP function **trim()** accomplish?

19. Briefly describe **SOAP**.

20. Using PHP, how can a text string be made safe for transmission as a URL parameter?

21. Name some weaknesses of **FTP**.

22. What is a recursive function?

23. What is **OOP**? State its major benefits.

24. Which of the major tenets of **OOP** are not natively supported by PHP?

25. Give a brief code example of object inheritance in PHP.

26. Does procedural programming have any advantages over **OOP**?

27. What are the hazards of using global variables?

28. Give a brief example of a db schema that is not normalized.

29. Now, normalize the above schema.

30. Write an SQL statement that modifies an existing record from the above normalized schema.

31. When would it make sense to decrease the degree of normalization?

32. What is **PLSQL** (or PLPGSQL, PLMYSQL …etc.) and when should it be used?

33. When combining PHP code with HTML, do you prefer to …
    a. Insert the PHP code into the HTML via PHP tags
    b. Output the HTML code from within your PHP tags?

34. Apart from your browser, which applications do you prefer to use to connect to a remote server?

35. Do you use an HTML editor? If so which?

36. Which text editor do you use when writing PHP code?

Conclusion

A proposed three-pronged solution was implemented.
1. Actively involve industry in order to better understand its dynamically changing needs.
2. Maintain knowledge of WSL usage, trends and growth.
3. Constantly seek opportunities to integrate WPL usage in instruction, homework, lab and project activities.

We determined that PHP would be a prime candidate for experimentation. Hence, it was incorporated into every facet of an Information Assurance and Security course of instruction as well as an Identity Mangement course of instruction. The additional web and security concepts introduced in the industry exam have helped focus on the more demanding aspects of building a secure WSL environment.

Acknowledgements

The participation of academicians from IT universities in the WSL survey is appreciated. Their input will be beneficial to all who want to produce quality courses of instruction. Additionally, the sharing and dialogue accomplished with our industry partners is noted and will be put to good use, hopefully, by preparing better future employees for their venture.

Bibliography

[1] http://en.wikipedia.org/wiki/Scripting_programming_languages March 7, 2006
[2] http://tiobe.com/tpci.htm March 7, 2006
[3] http://www.securityspace.com/s_survey/data/200602/index.html March 7, 2006

[4] http://www.securityspace.com/s_survey/data/man.200602/apachemods.html March 7, 2006
[5] http://lists.nyphp.org/pipermail/talk/2003-September/005231.html March 7, 2006
[6] http://www.php.net/usage.php March 7, 2006
[7] http://news.com.com/2100-7344_3-5589559.html March 7, 2006
[8] http://www.radwin.org/michael/blog/2003/02/one_year_of_php_at_yahoo.html March 7, 2006