Evaluation of a Web-Based Tutoring System for Java: Design Implications

Henry H. Emurian
UMBC, 1000 Hilltop Circle, Baltimore, MD 21250, USA emurian@umbc.edu

ABSTRACT

To assess the relative effectiveness of a Java tutor, three groups of eight students learned a Java applet under three instructional conditions: (1) web-based programmed instruction tutoring system, (2) self-regulated learning with a manual, and (3) rote memorization of the code. Post-learning multiple-choice tests were administered for 32 items of code, 10 rows of code, and 12 general principles of Java programming. It was hypothesized that these performance measures would show the superiority of the tutoring system, when compared to studying a manual of the syntax and semantics of the code and to memorizing the code without learning the meaning of the items in the applet. The results did not support test outcome differences among the three groups on these measures, and software self-efficacy improved for all students. These findings were interpreted in terms of implications for the design of tutoring system instructional frames and in terms of the sensitivity of the tests to detect differences among the three groups.

INTRODUCTION

For the past several years, we have developed a web-based tutoring system that teaches novice information systems students how to write and to understand a simple Java applet. The process of improving the system over successive semesters, based on the learning performance and usability ratings of students who used the system in the classroom, is similar to an action research perspective in the field of education, as explained by Elias and Dilworth (2003). Adjusting teaching strategies within the context of the classroom reflects a *design-based research methodology* (Hoadley, 2004).

We first reported the instructional system in Emurian, Hu, Wang, and Durham (2000). The next several years were devoted to improving the system and demonstrating its effectiveness to meet the needs of novice learners and its perceived value by our students (Emurian, 2004a; Emurian & Durham, 2001, 2002; Emurian, Wang, & Durham, 2000). We also showed that the tutor's instructional frames, which explained an item of Java code, led students to understand general principles that could be applied to new situations (Emurian, 2004b; Emurian, in press). The behavior analysis principles that underlie the tutoring system design, which is based on programmed instruction, are presented in Emurian and Durham (2003), Emurian, Wang, and Durham (2003), Greer (2002), and Greer and McDonough (1999).

Although systematic replication within the classroom was our methodological approach, the *gold standard* for providing the evidentiary base of instructional effectiveness is the *randomized field trial* (Towne & Hilton, 2004). The U.S. Department of Education maintains a public database of studies that are judged rigorous, in terms of experimental design and data analysis[2]. Disincentives exist, however, for faculty teaching in research universities to expend energy on instructional improvements (McCray, DeHaan, & Schuck, 2003), despite the growing concern that instructional tactics, at least within K-12 schools, be based on so-called scientific evidence of effectiveness (Viadero, 2004). The present study, then, is a first attempt to evaluate the tutor with a randomized trial.

We evaluated the tutoring system's effectiveness against performance by a second group of students who were given a manual of information to study that was similar to a chapter in a textbook. Because our experience indicated that patterns evident in the Java code carried information about program functionality, we included a third group of students who memorized the program but without information about syntax and semantics. This project intended to pilot learning conditions that might be investigated subsequently within a more comprehensive experiment.

METHOD

- **Subjects.** Undergraduate students in Information Systems were recruited by announcements in class and listserv postings. The only criterion for acceptance was no reported experience in Java. Subjects were compensated $25 for participation, and informed consent was obtained. Prior to the study, the subject completed a demographics questionnaire, which included a report of Java experience, and a software self-efficacy questionnaire based on the code in the program to be learned.
- **Experimental design.** This was a between-subjects randomized design with three treatment groups: (1) a Tutor group, (2) a Manual group, and (3) a Code group. Subjects were assigned to treatments by block randomization.
- **Procedure**. For the Tutor group, the subject completed the web-based tutoring system. The tutor taught a simple Java applet, which was organized into 32 items and ten rows of code. The tutoring system is freely accessible on the Web[3].

The final section in the online tutor, the Program Interface, required the subject to type the program into a text area input field. If there was an error, the subject viewed the correct program and tried to enter it again. That cycle repeated until the program was entered correctly. All three groups completed this last section. When the cycle was repeated, that event was counted as one error.

For the Manual group, the subject was presented with a paper version of all instructional material that was presented in the online tutor. The difference between the Tutor and the Manual groups was that the paper version omitted the multiple-choice tests for items and rows that were embedded within the online tutor. The paper version also did not have the interfaces that allowed the subject to practice typing the symbols in the program prior to learning the meaning of the items of code. The subject was instructed to study the manual until he or she indicated readiness (1) to be tested on the meaning of the items and rows of code presented and (2) to enter the code into the Program Interface. The subject was also informed that studying could last no longer than 2.5 hours. This limit was chosen because almost all students in our classes complete the tutoring system within that time.

The Code group was presented only with the final Program Interface. That group of subjects only memorized the code and typed it without being taught the meaning of the items and rows. For all three groups, the code had to be typed correctly in the Program Interface for the study to be concluded.

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Figure 1. Characteristics of Subjects

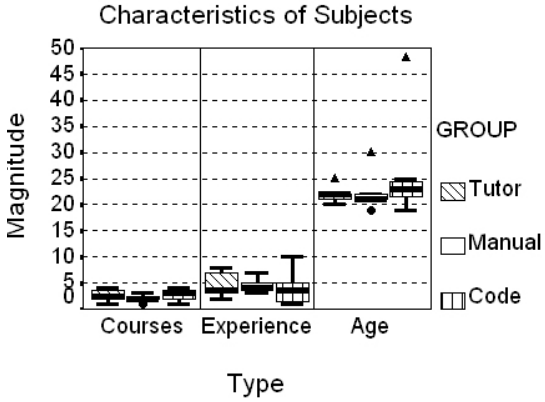
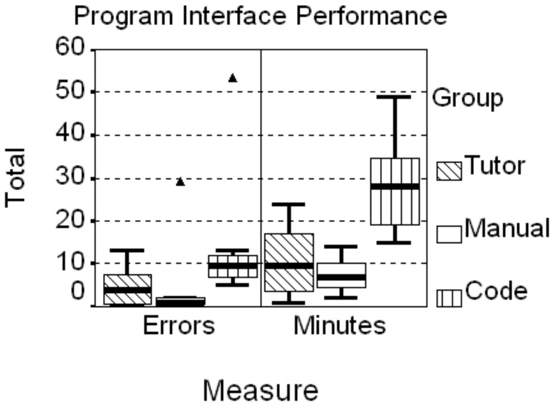


Figure 3. Program Interface Performance



RESULTS

Unless otherwise noted, the Kruskal-Wallis (K-W) *ANOVA by ranks* test was used because it is a conservative non-parametric test that is best applied to ordinal and ratio data with small sample sizes (Maxwell & Delaney, 2000, p. 703). The test is based upon a Chi-Square distribution.

Figure 1 presents boxplots of the subjects' reports of the number of prior programming *courses* taken, programming *experience*, and *age*. Programming experience was assessed by a 10-point ordinal scale where ***1 = No experience (novice)*** to ***10 = Extensive experience (expert)***. On a similar ordinal scale for Java experience, all 24 subjects reported a rating of one. The median number of prior programming courses that the subjects had taken was fewer than four for all three groups, and a K-W test was not significant (Chi-Square = 2.03, df = 2, p > .35). The median programming experience reported by the subjects was less than five for all three groups, and a K-W test was not significant (Chi-Square = 0.80, df = 2, p > .65). The median age reported by the subjects was less than 25 years for all three groups, and a K-W test was not significant (Chi-Square = 2.73, df = 2, p > .25). There was no evidence, then, that differences existed among the groups for the characteristics that were assessed.

Figure 2 presents boxplots of total errors observed on the multiple-choice tests for Java items, rows, and rules across the three groups. For items and rows in the Tutor group, data are presented for the multiple-choice tests that were embedded within the tutor (**Tutor-a**) and that were also administered as a post-tutor assessment (**Tutor-b**). For the Tutor group, a Friedman's test, which is appropriate for related samples, for item errors between these two occasions was significant (Chi-Square = 8.00, df = 1, p < .006), and a Friedman's test for row errors was also significant (Chi-Square =4.00, df = 1, p < .05). K-W tests for the three groups on the *post-learning assessments* were as follows for items (Chi-Square = 2.52, df = 2, p > .20), rows (Chi-Square = 1.33, df = 2, p > .50), and rules (Chi-Square = 0.31, df = 2, p > .80).

Figure 3 presents boxplots of errors committed and minutes taken on the final Program Interface across the three groups. For Errors, a K-W test was significant (Chi-Square = 8.30, df =2, p < .02). Since a pairwise test between the Tutor and Manual groups was not significant, those data were combined. A complex contrast between those combined groups and the Code group was significant[4] (Chi-Square = 7.69, df = 1, p < .008). For Minutes, a K-W test was significant (Chi-Square = 12.95, df = 2, p < .003). Since a pairwise test between the Tutor and Manual groups was not significant, those data were combined. A complex contrast between those combined groups and the Code group was significant (Chi-Square = 12.63, df = 1, p < .001).

Figure 4 presents boxplots of software self-efficacy ratings for pre-learning and post-learning occasions across the three groups. The scale was a ten-point ordinal scale where ***1 = No confidence*** and ***10 = Total confidence*** in being able to use the item of code. The median rating was higher for the post-learning occasion in comparison to the pre-learning occasion, and a Friedman's test was significant for the Tutor group (Chi-Square = 8.00, df = 1, p <.01), the Manual group (Chi-Square = 8.00, df = 1, p <.01), and the Code group (Chi-Square = 8.00, df = 1, p <.01). The median rating for the Manual group is graphically higher than the other

Figure 2. Errors on Multiple-Choice Tests

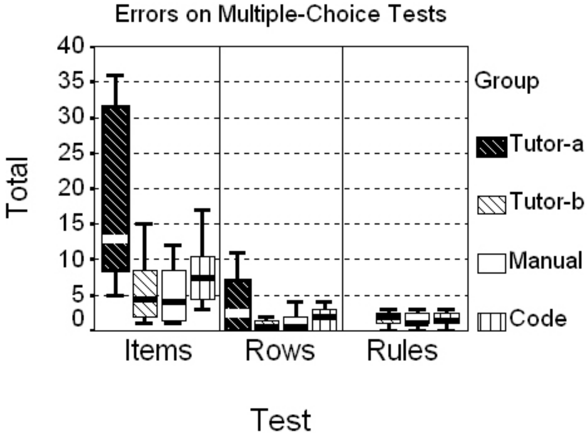
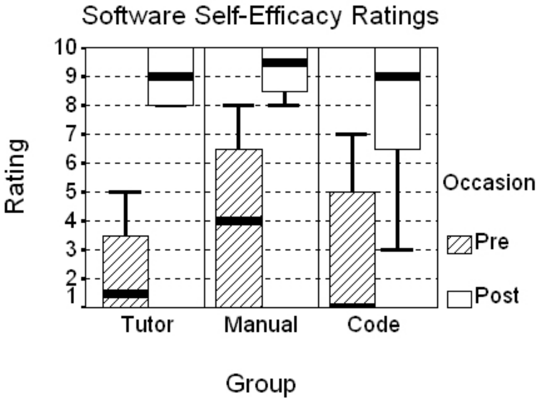


Figure 4. Software Self-Efficacy Ratings



Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

two groups for both occasions. However, a K-W test was not significant for either the pre-learning occasion (Chi-Square = 1.64, df = 2, p > .40) or the post-learning occasion (Chi-Square = 0.60, df = 2, p > .50).

DISCUSSION

The outcome of this experiment did not support the superiority of the web-based tutor in comparison to conditions where subjects were asked to study a manual in preparation for a test and where subjects only memorized the code. The only difference was the observation that the subjects in the Code group showed more errors and time on the Program Interface, in comparison to the other groups. The following discussion attempts to clarify and interpret these findings.

Studies that appear in the research literature are almost always those that report statistically significant effects of various treatments. Much can be learned, however, even when the null hypothesis can not be rejected. For example, this was demonstrated in a recent investigation of peer assisted tutoring (Rittschof & Griffin, 2001), which found no difference on test performance in an Education course among two peer tutoring conditions and an individualized study condition. Additionally, Buzhardt and Semb (2002) found no difference on final test performance among students who received initial test feedback under three different conditions: fixed item by item sequence, optional item by item sequence, and end of test.

In contrast, Gao and Lehman (2003) reported that students who used a web-based environment to learn about copyrights performed better on a post-learning achievement test in treatment groups that required answering multiple-choice tests and generating responses during the learning, in comparison to a group whose members only read the material. In the present study, the null hypothesis, which assumes that the data from the three groups could best be described by assuming that all subjects came from the same population, could not be rejected for the items, rows, and rules multiple-choice tests. It is informative to learn that self study may not always produce testing outcomes that differ from a structured approach to encountering new technical information, but it is clear that the literature is inconsistent in its findings, recommendations, and justifications for particular instructional design techniques.

The results, however, were notable in several ways. First, subjects using the tutoring system showed higher errors on the embedded multiple tests, in comparison to their own post-learning assessment and in comparison to the other two groups. This suggests that the subjects may have attempted to reduce *cognitive workload* (van Merrienboer, Kirschner, & Kester, 2003) by attending less closely to the textual information presented prior to a multiple-choice test, in comparison to the learners who had all tests only at the conclusion of their condition. Since each test for an item and a row in the tutor could be repeated until correct, this opportunity likely allowed more guessing to occur on the initial attempt to pass the test. What is surprising, however, was the failure of the tutor test learning to carry over (*i.e.*, transfer) to the same tests administered in paper form at the post-tutor assessment. That is, there was no difference in post-learning test performance between the Tutor and the Manual groups despite the fact that the tutor subjects had previously passed those test items correctly.

A second notable outcome was the observation that subjects who only memorized the Java code could not be differentiated, statistically, from the other subjects in terms of the items, rows, and rules tests. This was evident even though the post-learning medians for the Code group in Figure 2 are graphically higher for the items and rows tests. Although it may not be surprising that subjects within the Code group required more errors and minutes to type the program correctly, in comparison to the other groups, the outcomes for the *meaningful learning* (Mayer, 2002) rules test present a complex interpretative challenge.

In response to this outcome, several of the subjects in the Code group were contacted and asked about their surprising accuracy on the rules test, especially in light of the fact that those subjects had only memorized the code. Despite the fact that no subject had Java experience, subjects within the Code group reported that the patterns that were memorized provided *information* that was useful in selecting answers to the multiple-choice tests. Although these observations are unsystematic and anecdotal, they do shed light on the *value* of learning a program, despite criticisms of rote memorization in science, technology, and mathematics education (Bransford, Brown, & Cocking, 2000).

A third notable outcome was reflected in the software self-efficacy ratings, as shown in Figure 4. Consistent with our previous work, software self-efficacy improved for the Tutor group between pre-tutor and post-tutor occasions. Surprisingly, however, both the Manual and Code groups of subjects also showed such improvements, and post-learning differences among the three groups were not observed, even given the slightly elevated median for the Manual group observed during the pre-learning assessment. These are ordinal data, not ratio data, and that makes it problematic to interpret outcomes other than a change in a particular subject's rating from one occasion to another.

FUTURE DIRECTIONS

Although statistical *power* is obviously an issue with such small sample sizes as those used here, increasing the sample size to obtain a statistically significant outcome may have no practical value to educators unless the effect size is robust. Rather than advocating replication with more subjects, our strategy is to interpret the current outcomes with a view to potentiating the effectiveness of the web-based tutoring system.

Our first interpretation is that the tests embedded within the tutoring system were too easy to be sensitive to the various conditions that were investigated here. In response, the items, rows, and rules tests have been revised with the objective of making them challenging so that users of the system will read the textual frames more closely than done previously5. Second to be considered is to increase the size of the *learn unit* (Greer & McDonough, 1999) so that testing will only occur after a series of item frames, not following just a single frame. For example, the items might be clustered into rows for the purpose of testing. At the completion of a row in the items learning stage of the tutor, multiple-choice tests on each item in that row would be administered. An error on any individual item test would recycle the tutor to repeat the entire row of items, not just a single item. The impact of these modifications, suggested by the outcomes of this first experimental analysis, will require empirical validation.

Such a revision might occasion a learner's increase in motivated attention and self-regulated learning to a greater extent than occurs with the current design of the tutoring system. As stated by Woolfolk, Winne, and Perry (2000, p. 384), educators need to impart to students *a combination of academic learning skills and self-control that makes learning easier, so learners are more motivated; in other words, they have the skill and will to learn* (cited in Martin, 2004). The educational literature and the experiences of teachers indicate that there are many ways to achieve that outcome. The techniques presented and discussed here are in furtherance of reaching that objective for our students. Much additional work needs to be done, however, and this first and essential randomized trial has been helpful in revealing what that should be.

REFERENCES

Bransford, J.D., Brown, A.L., & Cocking, R.R. (2000). *How People Learn: Brain, Mind, Experience, and School* (expanded edition). Washington, D.C.: National Academy Press.

Buzhardt, J., & Semb, G.B. (2002). Item-by-item versus end-of-test feedback in a computer-based PSI course. *Journal of Behavioral Education*, *11*(2), 89-104.

Conover, W.J. (1971). *Practical Nonparametric Statistics.* New York, NY: John Wiley & Sons.

Elias, M.J., & Dilworth, J.E. (2003). Ecological/developmental theory, context-based best practice, and school-based action research: cornerstones of school psychology training and policy. *Journal of School Psychology*, *41*, 293-297.

Emurian, H.H. (2004a). A programmed instruction tutoring system for Java™ consideration of learning performance and software self-efficacy. *Computers in Human Behavior*, *20*(3), 423-459.

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Emurian, H.H. (2004b). Web-based tutoring for Java: evidence of rule-governed learning. In M. Khosrowpour (Ed.), Innovations Through Information Technology *(pp. 20-23)*. Hershey: Idea Group Publishing.

Emurian, H.H. (in press). Web-based programmed instruction: evidence of rule-governed behavior. *Computers in Human Behavior*.

Emurian, H.H., & Durham, A.G. (2001). A personalized system of instruction for teaching Java. In M. Khosrowpour (Ed.), *Managing Information Technology in a Global Economy* (pp. 155-160). Hershey: Idea Group Publishing.

Emurian, H.H., & Durham, A.G. (2002). Enhanced learning on a programmed instruction tutoring system for JAVA. In M. Khosrowpour (Ed.), *Issues and Trends of IT Management in Contemporary Organizations* (pp. 205-208). Hershey: Idea Group Publishing.

Emurian, H.H., & Durham, A.G. (2003). Computer-based tutoring systems: a behavioral approach. In J.A. Jacko & A. Sears (Eds.), *Handbook of Human-Computer Interaction* (pp. 677-697). Mahwah, NJ: Lawrence Erlbaum & Associates.

Emurian, H.H., Wang, J., & Durham, A.G. (2000). Design of a web-based tutoring system for Java. In M. Khosrowpour (Ed.), *Challenges of Information Technology Management in the 21st Century* (pp. 757-759). Hershey: Idea Group Publishing.

Emurian, H.H., Wang, J., & Durham, A.G. (2003). Analysis of learner performance on a tutoring system for Java. In McGill, T. (Ed.), *Current Issues in IT Education* (pp. 46-76). Hershey, PA: IRM Press.

Emurian, H.H., Hu, X., Wang, J., & Durham, A.G. (2000). Learning Java: a programmed instruction approach using applets. *Computers in Human Behavior*, *16*, 395-422.

Gao, T., & Lehman, J.D. (2003). The effects of different levels of interaction on the achievement and motivational perceptions of college students in a web-based learning environment. *Journal of Interactive Learning Research*, *14*(4), 367-386.

Greer, R.D. (2002). *Designing Teaching Strategies: An Applied Behavior Analysis Systems Approach*. New York, NY: Academic Press. Greer, R.D., & McDonough, S.H. (1999). Is the learn unit a fundamental measure of pedagogy? *The Behavior Analyst*, *22*, 5-16.

Hoadley, C.M. (2004). Methodological alignment in design-based research. *Educational Psychologist*, *39*, 203-212.

Martin, J. (2004). Self-regulated learning, social cognitive theory, and agency. *Educational Psychologist*, *39*(2), 135-145.

Maxwell, S.E., & Delaney, H.D. (2000). *Designing Experiments and Analyzing Data*. Mahwah, NJ: Lawrence Erlbaum Associates.

Mayer, R.E. (2002). *The Promise of Educational Psychology. Volume II. Teaching for Meaningful Learning*. Upper Saddle River, NJ: Pearson Education, Inc.

McCray, R.A., DeHaan, R., & Schuck, J.A. (Eds.) (2003). Improving Undergraduate Instruction in Science, Technology, Engineering, and Mathematics: Report of a Workshop. National Research Council. Steering Committee on Criteria and Benchmarks for Increased Learning from Undergraduate STEM Instruction. Committee on Undergraduate Science Education, Center for Education, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academies Press.

Rittschof, K.A., & Griffin, B.W. (2001). Reciprocal peer tutoring: re-examining the value of a co-operative learning technique to college students and instructors. *Educational Psychology*, *21*(3), 313-331.

Towne, L., & Hilton, M. (Eds.) (2004). *Implementing Randomized Field Trials in Education: Report of a Workshop*. Committee on Research in Education. Center for Education, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academies Press.

van Merrienboer, J.J.G., Kirschner, P.A., & Kester, L. (2003). Taking the load off a learner's mind: instructional design for complex learning. *Educational Psychologist*, *38*(1), 5-13.

Viadero, D. (2004). Education department issues practical guide to research-based practice. *Education Week*, *23*(16), 12.

Woolfolk, A.E., Winne, P.H., & Perry, N.E. (2000). *Educational Psycholog*y (Canadian edition). Scarborough, Ontario, Canada: Allyn and Bacon Canada.

ENDNOTES

[1] The author is indebted to Ms. Lidan Ha for managing this project.

[2] http://www.w-w-c.org/

[3] http://nasa1.ifsm.umbc.edu/learnJava/tutorLinks/TutorLinks.html

[4] Although interpretations of p values are problematic for post-hoc contrasts using non-parametric techniques (Conover, 1971), the ordering of the mean rankings supports the conclusion that errors were reliably higher for the Code group.

[5] The instructional manual and tests are available: http://nasa1.ifsm.umbc.edu/learnJava/savetext/TutorContent.pdf

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.