ABSTRACT

Students in a graduate class and an undergraduate class in Information Systems completed a Web-based programmed instruction tutor that taught a simple Java applet as the first technical training exercise in a computer programming course. The tutor is a competency-based instructional system for individualized distance learning. When a student completes the tutor, the student has achieved a targeted level of understanding the code and has written the code correctly from memory. Before and after using the tutor in the present study, students completed a software self-efficacy questionnaire and a test of the application of general Java principles (far transfer of learning). After completing the tutor, students in both classes showed increases in software self-efficacy and in correct answers on the test of general principles. These findings contribute to the stream of formative evaluations of the tutoring system. They show the capacity of the Web-based tutor to generate meaningful learning (i.e., understanding of concepts) at the level of the individual student.

Keywords: computer-assisted instruction; Java training; meaningful learning; programmed instruction

INTRODUCTION

This article presents a continuation of a series of formative evaluations to assess and to enhance the instructional effectiveness of an automated and individualized distance learning system that is intended to assist Information Systems students in beginning their study of Java™. We previously reported our progress in the development of this tutoring system, which teaches a simple Java applet, and its application as the first technical training exercise for students in a computer programming course (Emurian, 2004; Emurian & Durham, 2001, 2002; Emurian, Hu, Wang, & Durham, 2000). The purpose of the tutor is to provide each and every student with a documented and identical level of elementary knowledge and skill. The tutoring system has been demonstrably effective in promoting technical skill and in cultivating self-confidence in beginning students by giving them a successful learning experience that motivates their further study of Java using text-
books, lectures, laboratory demonstrations, independent problem solving, and the like. The tutoring system is targeted to Information Systems majors, whose primary interests may lie in systems analysis and design, database, decision support systems, knowledge management, and information resource management, but who would benefit from acquiring elementary skill in an object-oriented programming language such as Java.

One of the challenges of developing an automated distance learning system, however, is to craft the instructional experience so that students acquire the capability to solve problems not explicitly taught or encountered in the system itself. When students are able to apply knowledge successfully to new situations, they are said to be demonstrating meaningful learning (Mayer, 2002) as opposed to reciting facts acquired by rote memorization. These two outcomes reflect the opposite endpoints on a generality-specificity dimension of skill (Novick, 1990). Generalizable rules, which may be the essence of meaningful learning, can be acquired by direct instruction and rehearsal or by induction, when many different situations are encountered that exhibit the general rule (Kudadjie-Gyamfi & Rachlin, 2002). The former tactic is consistent with our instructional system design, which is competency-based and intended to insure that all students reach the same level of knowledge and skill with the applet being taught.

The theory supporting the development of the tutoring system is a behavior-analytic model based upon the learn unit formulation of Greer and McDonough (1999) as applied to programmed instruction for technology education (Greer, 2002). The interactive tutoring system interfaces that are presented to the learner reflect the systematic increase in the size of the learn units from simple symbol production (i.e., learning to type) to writing and understanding the entire program. The stream of work leading up to the present study constitutes a series of replications over which the system was refined to ensure that mastery would occur for all learners. This approach is similar to instructional material improvements suggested by formative evaluation (Harley, Seals, & Rosson, 1998) and by design-based research (Hoadley, 2004), and how this strategy contrasts with between-subjects evaluations is discussed in subsequent paragraphs.

The purpose of the present study is to show that students who complete the tutor do acquire general rules that are applicable to programming problems not explicitly addressed in the tutor itself. The study extends our prior investigation (Emurian, 2005) by increasing the number of testable rules to 10 and by supporting the reliability of the outcomes over two different groups of students. This research approach constitutes systematic replication (Sidman, 1960), which is an alternative to null hypothesis testing and intended to validate externally this particular instructional system design rather than to test hypotheses regarding effect sizes across alternative designs. This methodology falls within the scope of an outcomes assessment model of evaluating teaching effectiveness (Fox & Hackerman, 2003), and, in the present situation, the teacher is the Web-based tutoring system.

Interpretative surveys of the scientific literature in far transfer effects of learning continue to show the advantages of explicitly teaching generalizable principles and rules rather than expecting such knowledge to develop implicitly or abstractly as a byproduct of memorizing facts (Barnett & Ceci, 2002) or by pure discovery learning (Mayer, 2004). For example, it is likely more efficient to teach students the rule to begin the name of a Java class with a capital letter than to expect students to discover such a rule inductively by studying many different programs and by trying to discern commonalities among them. In fact, a combination of teaching rules with examples might be optimal for meaningful learning, and our approach to the design of the tutoring system is based on that latter assumption. The study to follow, then, assesses the extent of rule-governed knowledge before and after students have used the tutoring system.