The Effectiveness of Visualization for Learning Expression Evaluation

Amruth N. Kumar
Ramapo College of New Jersey
505 Ramapo Valley Road
Mahwah, NJ 07430, USA
201 684 7712
amruth@ramapo.edu

ABSTRACT
A controlled study was conducted to evaluate the effectiveness of providing visualization as part of feedback in a problem-solving software tutor on arithmetic expression evaluation. Data was collected over six semesters from multiple institutions. ANOVA analysis of the collected data was conducted in three stages. Statistically significant results include that visualization helped students learn more concepts; visualization did not improve the speed of learning; the benefits of visualization accrued primarily to less-prepared students; and visualization may affect different demographic subgroups differently. INCIDENTAL results include that there was no difference among demographic groups (male/female, traditional/underrepresented, Computer Science/non-CS) in the number of concepts learned using the software tutor, although some groups (female, underrepresented) were less-prepared before using the tutor than their counterparts, and some groups learned concepts with fewer practice problems (male, traditionally represented) than their counterparts. Concurrence of the results obtained whether the analysis was conducted based on pre-condition (need) or post-condition (benefit) of using the tutor strengthens the claims made as a result of this study.

Categories and Subject Descriptors
K.3.2 [Computer and Information Science Education]: Computer Science Education

General Terms
Measurement, Performance, Experimentation.

Keywords
Visualization; Evaluation.

1. INTRODUCTION
Software visualization includes algorithm visualization and program visualization. Whereas algorithm visualization deals with algorithms at a high level of abstraction, program visualization deals with programs at lower levels of abstraction [1]. The term program visualization is used to mean both static and dynamic visualization, the latter commonly referred to as program animation. Program visualization systems have also been classified as specialized versus generic systems: specialized systems visualize specific programming constructs whereas generic systems deal with entire programming languages [1]. The subject of the current study is static program visualization, specialized for expression evaluation.

Numerous systems have been developed over the years for software visualization. A meta-study of the effectiveness of algorithm visualization [2] found that in roughly half the studies reviewed in the meta-study, no significant difference was found between algorithm visualization and an alternative treatment provided to the students. A systematic review of generic program visualization and animation systems cataloged a mix of systems that were never evaluated, those that did not yield positive results, those whose positive results were not statistically significant and those with significant positive results [1]. Another survey of successful evaluations of visualization systems noted that about half the evaluations were only about usability; and a third of the evaluations were informal, “with little contribution to future improvements” [3].

Few of the specialized visualization systems, i.e., those built for specific programming constructs, have been evaluated. Among those that have been evaluated, one study found that animation was no more effective than text explanation for learning the semantics of C++ pointers [4]. Another study found that graphic visualization with text explanation was better than graphic visualization alone when learning expression evaluation [5]. Clearly, the effort to definitively establish whether and when visualization benefits learning is ongoing.

In this context, a controlled study was conducted to evaluate the effectiveness of providing static visualization as part of feedback in problem-solving software tutors on expression evaluation. The questions addressed in the study included whether and when visualization benefited student learning. This being the study of a specialized program visualization system, while the results may not be directly comparable with those of generic program visualization/animation systems or algorithm visualization systems, they would add to the accumulating evidence on whether and when visualization helps improve learning.

2. METHODOLOGY
2.1 The Software Tutor
For this study, a software tutor on arithmetic expression evaluation was used. The tutor presents expressions to the
student, has the student evaluate each expression one operator at
a time, grades the student’s answer and provides feedback. The
feedback includes whether the student’s answer is correct and
step-by-step explanation of the correct answer, which has been
shown to help students learn [6]. A snapshot of the tutor is
shown in Figure 1. In the figure, the student’s answer is shown
in the left panel. The feedback provided to the student is shown
in the right panel.

The tutor covers the following concepts: correct evaluation,
precedence, and associativity of five operators (addition,
subtraction, multiplication, division and remainder), parentheses, coercion, and errors associated with the operators
(e.g., divide-by-zero error).

The tutor is accessible over the web – students can use it
anytime, anywhere, and at their own convenience. It is part of
a suite of problem-solving tutors available for introductory
programming topics, available for free for educational use called
problets (problets.org).

2.2 Protocol
The software tutor administers pre-test-practice-post-test
protocol as follows:

- **Pretest** – During pretest, the tutor presents one problem
  per concept. If a student solves a problem correctly, the
  student is given credit for the corresponding concept. No
  feedback is provided to the student, and no more problems
  on the concept are presented to the student. On the other
  hand, if the student solves a problem incorrectly, feedback
  is presented to the student immediately after the student
  submits his/her solution to the problem. Additional
  problems are presented on the concept during the
  subsequent stages.

- **Adaptive practice** – During this stage, additional problems
  are presented to the student on only the concepts on which
  the student solved problems incorrectly during the pre-test.
  For each such concept, the student is presented multiple
  problems until the student masters the concept, i.e., solves
  at least 60% of the problems correctly. On each problem,
  the student receives feedback explaining the correct
  answer.

- **Post-test** – During this stage, the student is presented test
  problems on the concepts that the student mastered during
  adaptive practice.

- **Demographics** – Students are provided the option to
  identify their demographic information, including sex and
  race. Demographic information is solicited after the pre-
  test-practice-post-test protocol to avoid the effects of
  stereotype threat [7].

The pre-test-practice-post-test protocol was limited to 30
minutes and was administered back-to-back, entirely over the
web.

A concept is considered to have been learned if the student
solves the problem on that concept incorrectly during pre-test,
solves enough problems during adaptive practice to master the
concept, and solves the post-test problem on the concept
correctly. Since the entire pre-test-practice-post-test protocol
was limited to 30 minutes, not all the concepts on which a
student solves the problem incorrectly during pre-test turn into
learned concepts: the student may run out of time and never get
to solve practice problems or post-test problem on some
concepts.

The study was controlled, i.e., participant schools were divided
into control and test groups. The tutoring experience of control
and test groups was identical except for the feedback provided
to them after they incorrectly solved a problem during pre-test
and after every problem during practice. Control group received
only text feedback explaining each step in the evaluation of the
expression, e.g.,

10 / 6 returns 1
Since both the operands are integers, integer division is performed.
Any fraction in the result is discarded.

In addition to text feedback, test group also received graphic
feedback in the form of an underbrace spanning the operator (/ in
the above example) and operands (10 and 6 above), with the
intermediate result (1 above) drawn centered underneath the
underbrace. Both forms of feedback are shown in the right-hand
panel in Figure 1.

The graphic visualization explains the order of evaluation of
operators, but not concepts such as coercion and errors. So, it
was provided in addition to rather than instead of text
explanation. Such simultaneous presentation of the same
information in text and visual forms is recommended by dual
coding theory [8], which postulates that visual and verbal
information are processed differently, in separate channels, to
organize the information and create separate mental
representations, either of which can be used later to recall the
information.

2.3 Data Collection
Data was collected over six semesters: Fall 2011-Spring 2014.
The same version of the tutor was used in all six semesters. The
tutor was used by students in introductory programming courses
from multiple institutions, mostly baccalaureate institutions, but
also some community colleges and high schools.

The pre-test contained 16 problems. Since students could use
the tutor as often as they pleased, when a student used the tutor
multiple times, only the first attempt when the student solved all
16 pre-test problems was considered. If the student never solved
all 16 pre-test problems, the attempt with the most number of
pre-test problems solved was considered. In order to eliminate
trial or unserious attempts by students, only those records were
considered where students solved at least 10 pre-test problems.
After this winnowing, 1415 students remained in control group
and 1872 students in test group.

2.4 Data Analysis
A typical expression contains one or more operators. Students’
answers were graded by the software tutor as the number of
operators correctly evaluated by the student, divided by the total
number of operators in the expression. Therefore, the score on
each problem was normalized to 0 \(\rightarrow\) 1.0 regardless of the
number of operators in the expression.

Univariate ANOVA analysis was conducted for the following
dependent variables:

- Pre-test mean score per problem, calculated as the total
  pre-test score divided by the number of pre-test problems
solved – this normalized the variation in the number of problems solved by students (10-16);

- Number of concepts learned;
- Number of practice problems solved per learned concept, calculated as the number of practice problems solved, divided by the number of concepts learned, once again, to normalize the variation in the number of concepts learned by students.

The following fixed factors were used for analysis:

- Sex: male or female - participants were asked to identify their sex (biological notion) rather than their gender (social/cultural notion of man/woman) [9].
- Representation – traditionally represented (Caucasians and Asians [10]) versus underrepresented (the other racial groups, viz., Black/African American, Hispanic/Latino, Native American, Native Hawaiian/Pacific Islander and Other races)
- Discipline – Computer Science (CS) versus non-CS (all other majors, including Information Systems, Engineering, Other Sciences, Business, Arts, Social Sciences, Humanities and Other.)

When student records are complete, i.e., sex, race, and major of every student is known, a single multi-factor ANOVA analysis suffices, e.g., ANOVA of a dependent variable (e.g., concepts learned), with treatment, sex, representation, and discipline as multiple fixed factors. Since providing demographic data was optional, not all student records were complete. Carrying out a single multi-factor ANOVA analysis would have reduced the subject pool to only those student records that were complete, eliminating from consideration partial student records that could have contributed to analysis of some, but not all the independent factors (e.g., analysis by sex, but not discipline).

In order to maximize the insight offered by the collected data, ANOVA analysis of each dependent variable was carried out in three stages:

1. With treatment as the sole fixed factor: the entire subject pool was utilized; and only the main effect of treatment was analyzed;
2. With treatment and one demographic factor (e.g., sex): the subset of the subject pool that identified the demographic factor was utilized; and both the main effect of the demographic factor and interaction of the demographic factor with treatment were analyzed;
3. With treatment and all the demographic factors: the subset of the subject pool that identified all the demographic factors was utilized; and only interactions among treatment and/or more than one demographic factor were analyzed.

The sample size was the largest in the first stage, varied by the demographic factor in the second stage, and was the smallest in the third and final stage of analysis. If the student records were complete, results of the third stage would have subsumed the results of the first two stages.

Analysis was conducted for two subsets of the subject pool:

- Those who needed the tutor, i.e., scored 0.95 or fewer points per pre-test problem, the maximum score per problem being 1.0;
- Those who benefited from the tutor, i.e., learned one or more concepts by using the tutor.

3. RESULTS

3.1 Students who needed the tutor

**Treatment only:** Analysis of pre-test score per problem with treatment as the fixed factor yielded no significant main effect \[F(1,11813) = 0.959, p = 0.758\], i.e., there was no difference in the prior knowledge of the control and test groups as measured by the pre-test. So, any subsequent difference between the groups can be attributed to the use of the tutor.

Analysis of the concepts learned yielded a significant main effect for treatment \[F(1,1116) = 3.925, p = 0.048\]: control group students without visualization learned significantly fewer concepts \(1.596 \pm 0.098\) than test group students with visualization \(1.731 \pm 0.090\). In other words, visualization helped students learn more concepts. (All means are presented at 95% confidence levels.)

Students in both control and test conditions were separated into two groups: those who scored above average \(0.8118\) points per problem and those who scored average or below average on the pre-test. Analysis was repeated with both treatment and above average grouping as fixed factors. A marginally significant interaction was observed between treatment and above average grouping as shown in Table 1. The improvement from control to test condition was statistically significant only for the average or below average group \[F(1,536) = 4.23, p = 0.04\]: the benefits of visualization accrued primarily to less-prepared students.

<table>
<thead>
<tr>
<th>Grouping</th>
<th>Control</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average or below</td>
<td>2.062 (242)</td>
<td>2.302 (295)</td>
</tr>
<tr>
<td>Above average</td>
<td>1.173 (266)</td>
<td>1.194 (314)</td>
</tr>
</tbody>
</table>

Analysis of the number of practice problems solved per learned concept yielded no significant main effect for treatment \[F(1,1116) = 0.166, p = 0.683\]: visualization did not improve the speed of learning.

**Treatment and Sex:** Analysis of the pre-test score per problem yielded a marginally significant main effect for sex \[F(1,1721) = 3.092, p = 0.079\]: female students scored less \(0.804 \pm 0.013\) than male students \(0.817 \pm 0.008\), i.e., female students were less-prepared before using the tutor than male students.

Analysis of the concepts learned yielded no significant main effect for sex \[F(1,1066) = 0.067, p = 0.795\]: no significant difference was found in the number of concepts learned by female and male students.

Analysis of the practice problems per learned concept yielded a marginally significant main effect for sex \[F(1,1066) = 3.228, p = 0.073\]: male students learned with significantly fewer problems per concept \(3.213 \pm 0.173\) than female students.
(3.524 ± 0.291), i.e., male students learned faster than female students.

**Treatment and Representation:** Analysis of the pretest score per problem yielded a significant main effect for representation [F(1,1423) = 27.479, p < 0.001]: traditionally represented students scored more per problem (0.83 ± 0.008) than underrepresented students (0.785 ± 0.015), i.e., underrepresented students were less-prepared before using the tutor than traditionally represented students.

Analysis of the number of concepts learned yielded no significant main effect for representation [F(1,1897) = 0.095, p = 0.758]: no significant difference was found in the number of concepts learned by underrepresented and traditionally represented students.

Analysis of the practice problems per learned concept yielded a significant main effect for representation [F(1,1417) = 5.883, p = 0.015]: surprisingly, Computer Science majors scored significantly less (0.808 ± 0.013) than non-CS students (0.826 ± 0.009), i.e., Computer Science majors were less-prepared before using the tutor than non-CS majors. One explanation for this counter-intuitive result may be that the subject pool contained over 660 Engineering majors, who may have had better Math preparation than Computer Science majors.

Analysis of the number of concepts learned yielded no significant main effect for discipline [F(1,1895) = 0.572, p = 0.45]: no significant difference was found in the number of concepts learned by Computer Science and non-CS students.

Similarly, analysis of the problems solved per learned concept yielded no significant main effect for discipline [F(1,1895) = 0.185, p = 0.667]: no significant difference was found in the speed of learning of Computer Science and non-CS students.

**Treatment, Sex, Representation and Discipline:** Analysis of the number of concepts learned yielded a marginally significant interaction between treatment, sex and discipline [F(1,1842) = 2.834, p = 0.093], as shown in Table 2, where mean scores are followed by N in parentheses: whereas male CS students and female non-CS students learned more concepts with visualization than without, the vice-versa was true for female CS students, and no difference was observed between control and test treatments for male non-CS students. In other words, visualization may affect different demographic subgroups differently: if so, interactions among demographic groups and subgroups must be considered when evaluating the effectiveness of visualization. Among the differences in Table 2, only the difference for female non-CS students was statistically significant: t(172) = -2.506, p = 0.013. So, when visualization increased the number of concepts learned by the overall subject pool, it was primarily for non-CS female students.

| Table 2. Treatment x Sex x Discipline Interaction on the Concepts Learned |
|-----------------------------|-----|-----|
| Sex  | Discipline | Control | Test |
| Male | CS         | 1.568 (113) | 1.649 (112) |
|      | Non-CS     | 1.729 (165) | 1.731 (231) |
| Female | CS         | 1.469 (32) | 1.333 (20) |
|      | Non-CS     | 1.457 (84) | 2.024 (86) |

**3.2 Students who benefited from the tutor**

Most of the very same results were observed for students who benefited from the tutor as for students who needed the tutor. So, the discussion in this section is abbreviated, except where differences between the results of the two groups were observed. **Concurrence of the results obtained whether the analysis was conducted based on pre-condition (need) or post-condition (benefit) of using the tutor strengthens the claims made as a result of this study.**

**Treatment only:** There was no difference in the prior knowledge of the control and test groups [F(1,1203) = 0.757, p = 0.384].

Students with visualization learned more concepts (1.697 ± 0.085) than those without (1.569 ± 0.093) [F(1,1203) = 4.009, p = 0.045].

Even though the size of above-average group increased in both control (295, up from 266) and test conditions (362, up from 314), the improvement from control to test condition was still not statistically significant for the group. In other words, the benefits of visualization accrued primarily to less-prepared students.

Visualization did not improve the speed of learning measured as the number of practice problems solved per learned concept [F(1,1203) = 0.993, p = 0.760].

**Treatment and Sex:** Analysis of the pre-test score per problem yielded a significant main effect for sex [F(1,1147) = 5.141, p = 0.024]: female students scored less (0.789 ± 0.015) than male students (0.810 ± 0.009), i.e., female students were less-prepared before using the tutor than male students.

No significant difference was found in the number of concepts learned by female and male students [F(1,1147) = 0.298, p = 0.585].

Male students learned faster than female students (3.133 ± 0.163 versus 3.429 ± 0.274 problems per learned concept), the difference being marginally significant [F(1,1147) = 3.318, p = 0.069].

**Treatment and Representation:** Underrepresented students were less-prepared before using the tutor than traditionally represented students [F(1,164) = 12.415, p < 0.001].

No significant difference was found in the number of concepts learned by underrepresented and traditionally represented students [F(1,164) = 0.2, p = 0.655].

Traditionally represented students learned faster than underrepresented students (3.137 ± 0.173 versus 3.586 ± 0.33 problems per concept) [F(1,164) = 5.606, p = 0.018].
Visualization studies. As a case in point, the difference between the subject pool may significantly affect the outcome of with demographic subgroups.

Students who benefited from visualization. This suggests that differently. In this study, it was primarily female non-CS Visualization was found to affect different demographic groups effectiveness and broaden their appeal.

Treatment, Sex, Representation and Discipline: In analysis of concepts learned, once again, significant interaction was found between treatment, sex and discipline \( F(1,904) = 3.821, p = 0.051 \), as shown in Table 3. So, the effect of visualization varies with demographic subgroups. The demographic composition of the subject pool may significantly affect the outcome of visualization studies. As a case in point, the difference between control and test groups was statistically significant only for non-CS female students \( t(185) = -2.727, p = 0.007 \), i.e., the increase in the number of concepts learned due to visualization applied primarily to non-CS female students.

| Table 3. Treatment x Sex x Discipline Interaction on the Concepts Learned |
|----------------|----------------|----------------|
|                | Discipline |               | Control | Test  |
| Male           | CS         | 1.536 (121)   | 1.607 (121) |
|                | Non-CS     | 1.716 (180)   | 1.703 (245) |
| Female         | CS         | 1.567 (33)    | 1.321 (22)  |
|                | Non-CS     | 1.427 (90)    | 1.960 (93)  |


Figure 1. Snapshot of the Arithmetic Expression Evaluation Tutor