122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA

Making Value for Society

Paper ID #13274

Student Performance Improvement using Interactive Textbooks: A Three-University Cross-Semester Analysis

Dr. Alex Daniel Edgcomb, University of California, Riverside

Alex Edgcomb finished his PhD in computer science at UC Riverside in 2014. Alex has continued working as a research specialist at UC Riverside with his PhD advisor, studying the efficacy of web-native content for STEM education. Alex also works with Zyante, a startup that develops interactive, web-native textbooks in STEM.

Prof. Frank Vahid, University of California, Riverside

Frank Vahid is a Professor of Computer Science and Engineering at the Univ. of California, Riverside. His research interests include embedded systems design, and engineering education. He is a co-founder of zyBooks.com.

Prof. Roman Lysecky, University of Arizona

Roman Lysecky is an Associate Professor of Electrical and Computer Engineering at the University of Arizona. He received his B.S., M.S., and Ph.D. in Computer Science from the University of California, Riverside in 1999, 2000, and 2005, respectively. His research interests focus on embedded systems, with emphasis on runtime optimization, non-intrusive system observation methods for in-situ analysis of complex hardware and software behavior, data-adaptable system, and embedded system security. He was awarded the Outstanding Ph.D. Dissertation Award from the European Design and Automation Association (EDAA) in 2006 for New Directions in Embedded Systems. He received a CAREER award from the National Science Foundation in 2009 and four Best Paper Awards from the ACM/IEEE International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS), the ACM/IEEE Design Automation and Test in Europe Conference (DATE), the IEEE International Conference on Engineering of Computer-Based Systems (ECBS), and the International Conference on Mobile Ubiquitous Computing, Systems, Services (UBICOMM). He is an inventor on one US patent. He has coauthored five textbooks on VHDL, Verilog, C, C++, and Java programming. His recent textbooks, published with Zyante, utilize a web-native, interactive, and animated approach that has shown notable increases in student learning and course grades. He has also received multiple awards for Excellence at the Student Interface from the College of Engineering at the University of Arizona.

Prof. Andre Knoesen, University of California, Davis

André Knoesen is a Professor of Electrical and Computer Engineering at the University of California, Davis. He received a B.S. in Electronic Engineering from the University of Pretoria, South Africa in 1980, and an M.S. and Ph.D. in Electrical Engineering from Georgia Institute of Technology in 1982 and 1987, respectively. He is a Fellow of the Optical Society of America. He performs research in biosensors, materials, and their applications in electronic and optoelectronic sensor systems. He has published more than 100 research papers and holds 2 U.S. patents.

Prof. Rajeevan Amirtharajah, University of California, Davis

Rajeevan Amirtharajah received the S.B. and M.Eng. degrees in 1994, and the Ph.D. degree in 1999, all in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA. His doctoral work developed micropower DSP systems which scavenge energy from mechanical vibrations in their environment and use that energy to process information provided by embedded and wearable sensors. From 1999 to 2002, as a senior member of the technical staff at High Speed Solutions Corp., Hudson, MA, later a subsidiary of Intel Corporation, he helped create innovative high performance multidrop bus technologies using electromagnetic coupling and pulse-based modulated signaling. He worked as an ASIC and mixed-signal circuit design consultant at SMaL Camera Technologies, Cambridge, MA, in 2003.

In July 2003, he joined the Electrical and Computer Engineering department at the University of California, Davis, where he is currently an associate professor. His research interests include low power VLSI design for sensor applications, powering systems from ambient energy sources, and high performance circuit and interconnect design. He received the National Science Foundation CAREER award in 2006. He is an inventor on over twenty United States patents and is a member of IEEE, AAAS, and Sigma Xi. In the 2012-2013 academic year, he was a visiting scholar at the Berkeley Wireless Research Center.

Mary Lou Dorf, University of Michigan, Ann Arbor

Mary Lou Dorf received a B.S. from Alma College (1967). She received a M.S. in mathematics (1969) and the Ph.D. (1990) in systems engineering both from the University of Toledo. In Jan 2002, she joined the Electrical Engineering and Computer Science Department at the University of Michigan as a Lecturer. She has received multiple awards for her innovations in teaching and excellence in service. Her research interests include women in computer science and engineering education. Her current efforts are on increasing the number of women who declare computer science as a major

Student Performance Improvement using Interactive Textbooks: A Three-University Cross-Semester Analysis

Alex Edgcomb*, Frank Vahid*, Roman Lysecky°,
Andre Knoesen†, Rajeevan Amirtharajah†, and Mary Lou Dorf‡

*Dept. of Computer Science and Engineering, Univ. of California, Riverside. Also with zyBooks.com
°Dept. of Electrical and Computer Engineering, Univ. of Arizona
†Dept. of Electrical and Computer Engineering, Univ. of California, Davis
‡Dept. of Electrical Engineering and Computer Science, Univ. of Michigan

Abstract

We conducted studies to determine whether introducing an interactive textbook into introductory STEM (science, technology, engineering, math) courses can improve student grades. An interactive textbook has substantially less text than a traditional textbook, instead having numerous embedded question sets designed for learning and not quizzing, numerous animations of key concepts, and some built-in tools. The studies included four introductory computer-programming courses at three research universities: C++ at Univ. of Michigan, C/C++ at Univ. of Arizona, and MATLAB at Univ. of California at Davis. For each, two course offerings were compared, the first using a static textbook or static web resources, and the second using an interactive textbook. Most other course features remained the same between offerings, including the teacher and the semester offering (e.g., Spring semester), or were very similar; differences are noted. We analyzed final course grades and performance on specific course items using nonparametric analysis with conservative Bonferroni adjustment for multiple tests. Across all courses involving 1,945 students, course grades improved by 0.28 points on a 0-4 point scale (p-value < 0.001), or 1/4 letter grade, from static to interactive textbook. More importantly, students in the lower quartile of the course improved by 0.38 points (p-value < 0.001), or 1/3 letter grade. Exam scores improved by 13.6% (p-value < 0.001) and project scores by 7.4% (p-value < 0.001) from static to interactive textbooks. 98% of the students subscribed to the interactive textbook and completed at least some activities on the interactive textbook. The average student completed 87% of the assigned activities in the interactive textbook.

Keywords: Online textbooks, static content, web-native content, interactive content, programming, computer science, computer engineering, digital learning, digitally-enhanced education.

Introduction

Student acquisition rates of required college textbooks continues to drop[11] with 65% of students at least once not purchasing a required textbook[19]. Possible reasons include cost, and a growing disconnect between traditional textbooks (even in electronic form) and current student learning styles involving media/interaction. For example, 22% of students do not purchase the required textbook[24]. Noting high costs and low usage rates, some teachers have stopped requiring textbooks, relying instead on various sources like miscellaneous web resources and instructor notes. Meanwhile, attrition rates in introductory STEM (science, technology, engineering, and math) courses are high, with a U.S. national average of 48% of bachelor's degree students leaving the STEM majors (and 69% for associate's degree students)[3].

We hypothesize that interactive textbooks can help counter these negative trends, leading to improved grades in introductory STEM courses. In this paper, "interactive textbook" refers to material involving less text, and instead having extensive learning-focused question sets, animations of key concepts, and interactive tools. Such an interactive textbook typically is written from scratch for the web, but could include a previously-static textbook that is thoroughly redone for the web. This paper specifically does not consider typical conversions of static textbooks into electronic form, even if supplemented with some interactive activities, as an interactive textbook; such materials commonly involve content bloat, and the existence of the large static portion does not address the disconnect with student learning styles.

A randomized control trial by Edgcomb[4] demonstrated significant learning outcome improvements for a particular topic in introductory programming, with 16% average improvements, and with 64% improvement for the initially weakest lower quartile. The comparison was between an electronic version of a textbook versus interactive learning material (both accompanied by a web-based programming environment for practice). Furthermore, students voluntarily spent twice as much time engaged in the interactive learning material, and also self-reported higher engagement.

Ultimately, however, the question remains whether such material leads to improved performance in a course. Answering that question can be difficult due to the many variables that differ across offerings of a course in different semesters.

This paper presents cross-semester case studies for four large introductory programming courses at three universities. Each course previously used a traditional textbook or no textbook, and each course then introduced a required interactive textbook as the main difference between classes (a class is a particular offering of a course). Other course features remained mostly the same, including the teacher; differences are noted in each case study. The analyses of the student grade performance in each case study, and a combined analysis, show significant grade improvements after introducing the interactive textbook.

Background

Interactive learning materials have been shown to improve student learning outcomes in computer science[4] and other STEM fields[2][9][26]. Edgcomb[4] conducted a controlled study with 136 participants comparing the lesson effectiveness of electronic textbooks having static web content versus interactive web-native content. Lesson effectiveness was measured with a pre- and post-lesson quiz. Students assigned the interactive content improved 16% more (p-value = 0.016) than students assigned the static content. More importantly, for the students starting in the lower-quartile, students assigned the interactive content improved 64% more (p-value < 0.001) than students assigned the static content. Students reported higher learning-engagement (p-value = 0.055) with the interactive content (5.3/6 points) than the static content (4.8/6 points). Also importantly, students chose to spend nearly double the amount of time (p-value < 0.001) with the interactive content (17.5 minutes) than the static content (9.4 minutes). Hagerty[9] showed that ALEKS, a web-based education system for college algebra, significantly improved student exam scores in college algebra courses.

Woody[27] conducted a study comparing a traditional textbook with an e-textbook, reporting student preference and self-reported usage. The e-textbook was not a web-native textbook, instead the e-textbook was a digital version of the traditional textbook. Woody concluded that, "the design of an e-book may need to differ from that of a textbook to make for a more constructive user experience". The participants selected for the study did not necessarily know how to study from an e-textbook, as the selected participants only had to have had the option to an e-textbook in a previous course. Further, student performance in the class was not measured, so the relative usefulness of traditional textbooks and e-textbooks was unclear.

Academic groups[14][17][18] and companies[1][5][6][10][25][28] are actively developing web-native textbooks. Learning with Python[14], developed by professors at Luther College, integrates a programming environment inside the textbook, and includes multiple choice and short answer questions, a code visualization tool, and videos. The Open Learning Initiative (OLI) at Carnegie Mellon University[18] offers 21 free courses with a web-native textbook, including videos, multiple choice questions, and interactive activities that are recorded. Inkling's[10] textbooks utilize hyperlinks and a note-taking platform alongside the textbook. For Dummies eLearning[6] includes multiple choice quizzes at the end of sections. Flat World Knowledge[5] develops textbooks that include re-orderable content, tool-tip definitions for key terms, and videos. Zyante's zyBooks[28] are web-native textbooks that include frequent animations, interactive tools, and a diverse array of interactive question sets. zyBooks are re-orderable, record student activity, and contain a student and instructor dashboard for monitoring activity.

Researchers have developed interactive web-native activities to explain data structures and algorithms[8][12][13][15][23]. VisuAlgo[23], developed at the National University of Singapore, is a collection of interactive, visual tools for algorithms and data structures. Similarly, Galles[8], at the University of San Francisco, has developed many interactive, visual tools from early data structures through undergraduate algorithms.

Interactive textbooks may benefit flipped and active learning classroom models that have been shown to improve student performance[7][16][20][21][22]. Freeman[7] metaanalyzed 225 studies comparing active learning and traditional lecture classes in undergraduate science, technology, engineering, and math (STEM). Active learning students had 6% higher exam scores than traditional lecture students. Also, a given student was 1.5 times more likely to fail when in a traditional lecture class than active learning class. Porter[16] found that across more than 10,000 students in a peer-instruction (PI) computer science course had a 67% lower fail rate (p-value < 0.01) than students in a traditional course. In similar work, Simon[20] found that students in a PI class experienced better class enjoyment, improved attendance, better attentiveness, and improved awareness of ones own learning. One criticism of flipped and active classrooms is that more time is focused on less material. Interactive textbooks are well-suited to before-class assigned reading as shown in this paper by the 87% average assigned reading completion.

Case study methods

Two classes of a given course were selected for comparison. The selection criteria included that both classes were taught:

- by the same instructor
- during the same semester (or quarter) of different years, e.g., Spring 2012 and Spring 2013
- within 2 years of each other

The two classes were evaluated for student performance on multiple class assessments, such as assignments, projects, exams, class scores, and class grades. Each assessment was compared between the classes using one measurement, the average student performance.

A Mann-Whitney U test was used to compare differences between classes because the data from class assessments were nonparametric. Interpretation of significance values was conservative, applying the Bonferroni correction for multiple tests. P-values denoted with * are interpreted as statistically significant.

The grade scales are omitted from this paper to avoid potential issues with students learning the exact grade cutoffs.

Case study 1: CS2 with C language

Class information

The course was ECE 275, "Computer Programming II", a sophomore-level course at the University of Arizona. The course teaches the C/C++ programming language.

The comparison is between the Fall 2012 semester using a traditional textbook, and the Fall 2013 semester using an interactive textbook[28] with 1411 activities. All other aspects of the course were mostly unchanged.

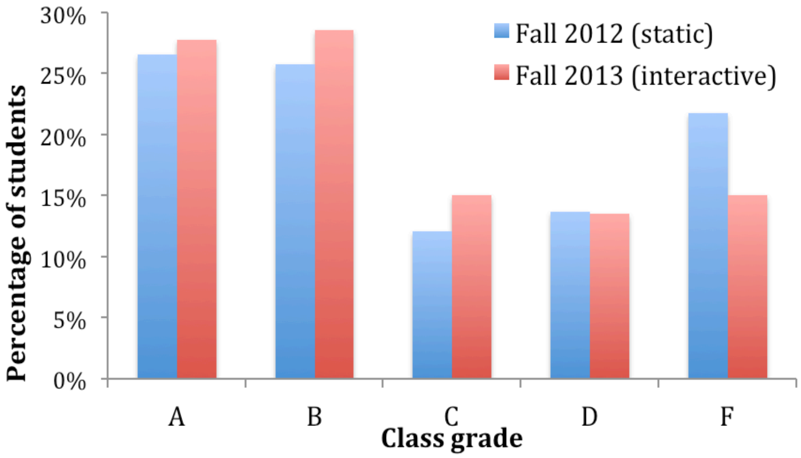
Fall 2012 had 124 students, and Fall 2013 had 133 students. Grades for both semesters were primarily from 40 points for programming assignments and 40 points for exams, of similar difficulty. Fall 2012 had 10 points for quizzes and 10 points for in-class and online exercises (instructor created). Fall 2013 had 10 points for quizzes and in-class-exercises, and 10 points for online exercises (interactive textbook activities, automatically logged when student does the activities while reading).

All of the students in Fall 2013 subscribed to the interactive textbook and completed at least one online exercise. The average student completed 87% of the assigned online exercises.

Analysis

The programming assignments average was 26.7 points for Fall 2012 and 29.6 points for Fall 2013. The exams average was 31.6 points for Fall 2012 and 30.4 points for Fall 2013. For Fall 2012, the quizzes average was 7.7 points and the in-class and online exercises average was 8.3 points. For Fall 2013, the quizzes and in-class-exercises average was 8.7 points and the online exercises a verage was 8.1 points. From Fall 2012 to 2013, assignment scores increased by 11.1% (p-value = 0.160). Exam scores were not significantly different (-3.6%, p-value = 0.512).

Figure 1. Case study 1: Grade distribution for Fall 2012 and 2013. Fall 2013 had more As, Bs, and Cs, and fewer Fs.



As shown in Figure 1, the Fall 2012 class had 27% As, 26% Bs, 12% Cs, 14% Ds, and 22% Fs. The Fall 2013 class had 28% As, 29% Bs, 15% Cs, 14% Ds, and 15% Fs. On a 0 to 4 point grade scale, the students using the interactive textbook averaged 0.19 higher final grade (p-value = 0.369) than the students using the traditional textbook.

More importantly, for the lower half of the class, the students using the interactive textbook had 0.34 grade points higher (p-value = 0.059) than the students using the traditional textbook, on a 0 to 4 scale, meaning an increase of 1/3rd a letter grade. For the lowest quartile of the class, the students using the interactive textbook averaged 0.26 higher grade points (p-value = 0.017) than the students using the traditional textbook.

In Fall 2012, 97 out of the 124 students passed, meaning earned higher than an F, (78.2% pass rate), while in Fall 2013, 113 out of the 133 students passed (85.0% pass rate). Proportionally, 8.6% more students passed (p-value = 0.164) the class using the interactive textbook than the class using the traditional textbook.

* Note: The Bonferroni correction for this section requires a p-value < 0.008 to be interpreted as significant.

Case study 2: Problem solving with MATLAB

Class information

The course was ENG 06, "Engineering Problem Solving", a freshmen-level course at the University of California at Davis. The course uses Matlab for engineering problem solving.

The comparison is between the Winter 2012 quarter requiring a static textbook plus additional instructor-provided notes, and the Winter 2014 quarter requiring an interactive textbook[28] with 861 activities. Winter 2012 was co-taught by two instructors, one of which was the sole instructor for Winter 2014. The instructors collaboratively developed the class materials. All other aspects of the class were mostly unchanged.

Winter 2012 had 211 students, and Winter 2014 had 245 students. Grades for both quarters were from 4 categories (worth 100 points each): weekly assignments (homework and lab), project #1 (individual), project #2 (team), and final exam. The final exam was multiple choice for both Winter 2012 and 2014, using the same questions in a shuffled order. For Winter 2014, the weekly assignments included online exercises (interactive textbook activities, automatically logged when student does the activities while reading) for 25% of that category's grade.

During Winter 2012 and 2014, the weekly assignments assessed the student's knowledge for that week's material. The lab sections contained under 30 students with the format: review a particular aspect of the material, then a 20 minute quiz. The interactive textbook was used to ensure that the student received feedback in preparation for the quiz in lab.

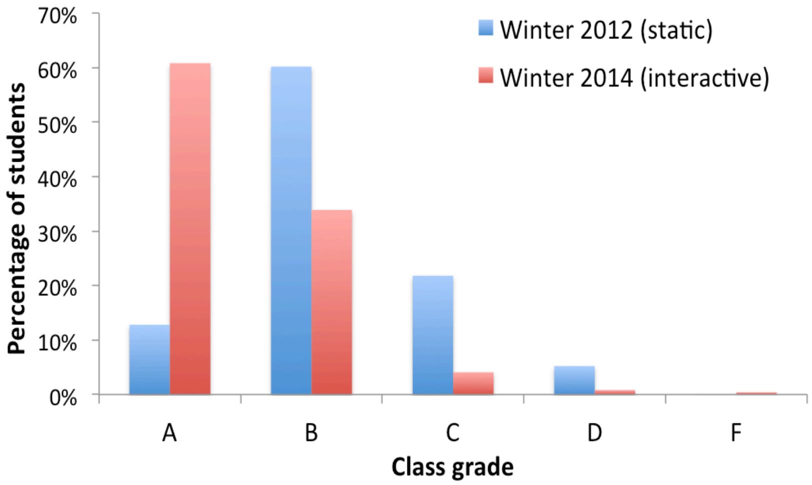
The categories were weighted as follows: weekly assignments worth 40% of overall grade for Winter 2012 and 2014, project #1 worth 5% for Winter 2012 and 10% for Winter 2014, project #2 worth 25% for Winter 2012 and 20% for Winter 2014, and final exam worth 30% for Winter 2012 and Winter 2014.

Analysis

The weekly assignments average was 75 points for Winter 2012 and 91 points for Winter 2014. The project #1 average was 81 for Winter 2012 and 81 for Winter 2014. The project #2 average was 89 for Winter 2012 and 92 for Winter 2014. The final exam average was 60 for Winter 2012 and 77 for Winter 2014. From Winter 2012 to 2014, weekly assignment scores increased by 17.1% (p-value < 0.001*), project #1 decreased by 0.7% (p-value = 0.031), project #2 increased by 3.7% (p-value < 0.001*), and final exam increased by 21.4% (p-value < 0.001*).

As shown in Figure 2, the Winter 2012 class had 12.8% As, 60.2% Bs, 21.8% Cs, 5.2% Ds, and 0.0% Fs. The Winter 2014 class had 60.8% As, 33.9% Bs, 4.1% Cs, 0.8% Ds, and 0.4% Fs. On a 0 to 4 point grade scale, the students using the interactive textbook averaged 0.70 higher final grade (p-value < 0.001*) than the students using the traditional textbook. From Figure 2, Winter 2014 had more As and fewer Bs, Cs, and Ds than Winter 2012.

Figure 2. Case study 2: Grade distribution for Winter 2012 and 2014. Winter 2014 had more As and fewer Bs, Cs, and Ds.



For the lower half of the class, the students using the interactive textbook had 0.72 grade points higher (p-value < 0.001*) than the students using the traditional textbook, on a 0 to 4 scale.

More importantly, for the lowest quartile of the class, the students using the interactive textbook averaged 0.93 higher grade points (p-value < 0.001*), nearly a full letter grade improvement, than the students using the traditional textbook.

* Note: The Bonferroni correction for this section requires a p-value < 0.007 to be interpreted as significant.

Case study 3: CS1 with C++ language

Course information

The course was EECS 183, "Elementary Programming Concepts" at the University of Michigan. The students in the courses were mostly sophomores (43% in Winter 2013 and 45% in Winter 2014) and juniors (28% in Winter 2013 and 26% in Winter 2014), and were primarily non-engineering (92% in Winter 2013 and 92% Winter 2014).

The comparison is between the Winter 2013 semester using static web resources, namely cplusplus.com, and the Winter 2014 semester using an interactive textbook[28] with 1409 activities. Both semesters also used an online auto-grading homework system. Both semesters taught C++. Winter 2014 added 3 weeks of Python, skipping C++ structs and moving exam reviews to Sunday evenings. All other aspects of the course were mostly unchanged.

Winter 2013 started with 434 students, and Winter 2014 started with 567 students. 35 students (8% of Winter 2013 students) withdrew from Winter 2013. 40 students (7% of Winter 2014 students) withdrew from Winter 2014. This analysis considers the 399 students from Winter 2013 and 527 students from Winter 2014 who did not withdraw.

Grades for both semesters were primarily from 6 programming assignments (350 points of Winter 2013 grade; 360 points of Winter 2014) and 3 exams (573 points of Winter 2013 grade; 560 points of Winter 2014), of similar difficulty. The instructor reported that programming assignment 4 was harder in Winter 2014, whereas programming assignment 6 was harder in Winter 2013. The remaining 80 points consisted of homework and participation for Winter 2013 and 2014. For Winter 2014, 25 points from homework and participation were for completing online textbook exercises (interactive book activities, automatically logged when student does the activities while reading).

98% of students in Winter 2014 subscribed to the interactive textbook and completed enough activities to earn points.

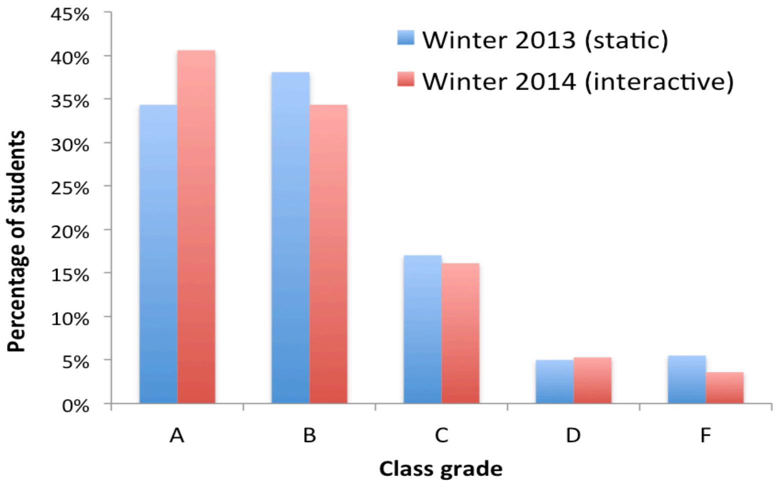
Analysis

The total points average for Winter 2013 was 823 out of 1003 points and Winter 2014 was 842 out of 1000 points. The programming assignment grade average for Winter 2013 was 313 out of 350 points and Winter 2013 was 328 out of 360 points. The exam grade average for Winter 2013 was 439 out of 573 points and Winter 2014 was 440 out of 560 points. The homework and participation grade average for Winter 2013 and Winter 2014 was 74 points.

The total points average improved by 2.3% (p-value = 0.398) from Winter 2013 to Winter 2014. The programming assignment grade average improved by 5.0% (p-value < 0.001*) from Winter 2013 to Winter 2014. The exam grade average improved by 2.5% (p-value = 0.228) from Winter 2013 to Winter 2014. The homework and participation average was the same between courses.

The Winter 2013 class had 34.3% As, 38.1% Bs, 17.0% Cs, 5.0% Ds, and 5.5% Fs. The Winter 2014 class had 40.6% As, 34.3% Bs, 16.1% Cs, 5.3% Ds, and 3.6% Fs. On a 0 to 4 point grade scale, the students using the interactive textbook averaged 0.12 higher final grade (p-value = 0.088) than the students using static web resources. From Figure 3, Winter 2014 had more As and fewer Bs and Fs.

Figure 3. Case study 3: Grade distribution for Winter 2013 and 2014. Winter 2014 had more As and fewer Bs and Fs.



For the lower half of the class, the students using the interactive textbook had 0.15 higher final grade (p-value = 0.070) than the students using static web resources, on a 0 to 4 scale, including 5.0% improvement, from 377 to 387, (p-value = 0.022) in exam scores and 8.3% improvement, from 279 to 302, (p-value < 0.020) in project scores.

More importantly, for the lowest quartile of the class, the students using the interactive textbook averaged 0.20 higher grade points (p-value = 0.064) than the students using static web resources, on a 0 to 4 scale, including 10.1% improvement, from 328 to 353, (p-value = 0.001*) in exam scores and 16.9% improvement, from 231 to 279, (p-value = 0.020) in project scores.

* Note: The Bonferroni correction for this section requires a p-value < 0.005 to be interpreted as significant.

Case study 4: CS1 with C language

Course information

The course was ECE 175, "Computer Programming I", at the University of Arizona. The course teaches the C programming language. The majority of students were engineers (60.9%), followed by science (19.4%) and letters arts (12.9%). The students were 16.6% Freshmen, 41.9% Sophomore, 27.4% Juniors, and 13.9% Seniors.

The comparison is between the Fall 2012 semester using a traditional textbook, and the Fall 2013 semester using an interactive textbook[28] with 627 activities. All other aspects of the course were mostly unchanged.

Fall 2012 had 140 students, and Fall 2013 had 166 students. Grades for both semesters were out of 100 points. Grades for both semesters were from two midterms (50 points total in Fall 2012; 30 points total in Fall 2013), a final project (25 points for Fall 2012 and 2013), homework (17 points for Fall 2012; 27 points for Fall 2013), and labs (8 points for Fall 2012; 9 points for Fall 2013). Fall 2012 also had extra credit points. Fall 2013 also had 9 points for online exercises (interactive textbook activities, automatically logged when student does the activities while reading).

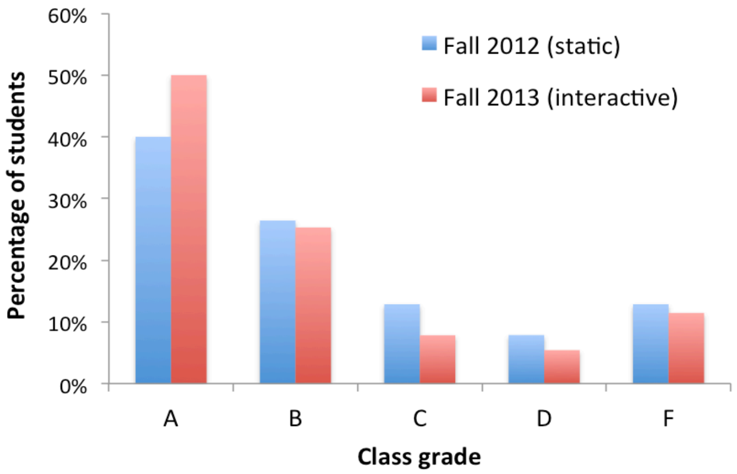
98.8% of students in Fall 2013 subscribed to the interactive textbook and completed at least one online exercise. The average student completed 88% of the assigned activities.

Analysis

The total points average for Fall 2012 was 79.2 points and Fall 2013 was 81.6 points. The midterm grade average for Fall 2012 was 35.0 out of 50 points, and for Fall 2013 was 22.3 out of 30 points. The final project average for Fall 2012 was 20.9 out of 25 points, and for Fall 2013 was 21.3 out of 25 points. The homework average for Fall 2012 was 14.4 out of 17 points, and for Fall 2013 was 21.8 out of 27 points. The lab average for Fall 2012 was 7.5 out of 8 points, and for Fall 2013 was 8.2 out of 9 points. The extra credit average for Fall 2012 was 1.4 points. The online exercise average for Fall 2013 was 7.9 out of 9 points.

The total points average improved by 3.0% (p-value = 0.121) from Fall 2012 to Fall 2013. The midterm grade average improved by 6.5% (p-value < 0.001*) from Fall 2012 to Fall 2013. The final project average improved by 2.0% (p-value = 0.001*) from Fall 2012 to Fall 2013. The homework average decreased by 4.9% (p-value < 0.001*), and labs decreased by 2.8% (p-value < 0.001*) from Fall 2012 to Fall 2013.

Figure 4. Case study 4: Grade distribution for Fall 2012 and 2013. Fall 2013 had more As and fewer Bs, Cs, Ds, and Fs.



The Fall 2012 class had 40.0% As, 26.4% Bs, 12.9% Cs, 7.9% Ds, and 12.9% Fs. The Fall 2013 class had 50.0% As, 25.3% Bs, 7.8% Cs, 5.4% Ds, and 11.4% Fs. On a 0 to 4 point grade scale, the students using the interactive textbook averaged 0.24 higher final grade (p-value = 0.072) than the students using static web resources. From Figure 4, Fall 2013 had more As and fewer Bs, Cs, Ds, and Fs.

For the lower half of the class, the students using the interactive textbook had 0.28 grade points higher (p-value = 0.098) than the students using the traditional textbook, on a 0 to 4 scale. For the lowest quartile of the class, the students using the interactive textbook averaged 0.20 higher grade points (p-value = 0.360) than the students using the traditional textbook.

* Note: The Bonferroni correction for this section requires a p-value < 0.007 to be interpreted as significant.

Cross-semester combined analysis

We conducted an analysis combining the 1,945 students in four courses. Five student assessments were consistent in each course and considered in the combined analysis: exams, projects, class score, class letter grade, and pass rate. For each student's assessment score, that score was Z-scored across both semester's of that course. The Z-scoring helps account for grading scales differences between courses.

Then, the Z-scores were concatenated for classes using static content, and separately concatenated for classes using interactive content. Finally, the concatenations were compared. Any student with a Z-score over 10 or under -10 were considered an outlier in the course and excluded from this analysis.

Significance was computed with the Mann-Whitney U test with two-tails because the assessments were not normal distributions. Also, a Bonferroni correction was applied to the interpretation of the significance value due to multiple tests, so a p-value < 0.01 is interpreted as statistically significant.

Table 1 shows the comparison between the static and interactive class Z-scores. The overall average Z-score is 0, so any assessment above 0 is above overall average. Similarly, any assessment below 0 is below overall average. The Z-score improvement is the difference between the interactive and static average Z-score. The improvement percentage is the percentage of area under a normal distribution for the Z-score improvement.

Table 1. Static and interactive classes compared across 5 assessments. The Z-score improvement is interactive Z-score average minus static Z-score average. Higher is better.

	Exam	**Projects**	**Class score**	**Class letter grade**	**Pass rate**
Static Z-score average	-0.192	-0.103	-0.228	-0.177	-0.029
Interactive Z-score average	0.157	0.084	0.140	0.144	0.044
Z-score improvement	0.349	0.186	0.368	0.321	0.073
Improvement percentage	13.6%	7.4%	14.3%	12.6%	2.9%
p-value	< 0.001	< 0.001	< 0.001	< 0.001	0.306

All of the interactive averages were above the respective static average. The exam average improved by 13.6% (p-value < 0.001) from static to interactive classes, projects average improved by 7.4% (p-value < 0.001), class score improved by 14.3% (p-value < 0.001), class letter grade improved by 0.28 points on a 4-point scale (12.6% improvement, p-value < 0.001). The lower quartile of students improved by 0.38 points on a 4-point scale (p-value < 0.001). The pass rate improved by 2.9% (p-value = 0.306). Note that the pass rate's p-value was larger than the other assessments because the number of failing students was a very small percentage of the total students.

Teaching participants

Roman Lysecky (ECE 275) is an associate professor at the Univ. of Arizona, and has taught the course 3 times.

Andre Knoesen (ENG 06) is a professor at the Univ. of California at Davis, and has taught the course 9 times.

Rajeevan Amirtharajah (ENG 06) is an associate professor at the Univ. of California at Davis, and has taught the course 3 times as of Fall 2014.

Mary Lou Dorf (EECS 183) is a Lecturer IV at the Univ. of Michigan, and has taught the course 24 times as of Fall 2014.

Loukas Lazos (ECE 175) is an associate professor at the Univ. of Arizona, and has taught the course 5 times.

The teaching participants were asked their opinion for what caused student performance improvement during the course offering using an interactive textbook. A common answer was that the interactive textbook helped better prepare students, such as, "I assigned [interactive textbook] readings due every two weeks. I noticed a difference. The students would ask more involved questions. They noticed that the reading did the basics and that lecture reviewed the basics, but went more in depth -- covering the exceptions and unusual cases." Another instructor wrote, "students read before class. Following on that, the questions students asked in lecture seemed to be less about syntax and more about design and high-level concepts."

The analyses were carried out by paper authors Edgcomb, a post-doctoral researcher at the Univ. of California, Riverside, and Vahid, a professor at Univ. of California, Riverside. Edgcomb is a developer at, and Vahid a co-founder of, Zyante, the publisher of the interactive books used in the study. Lysecky, Knoesen, and Amirtharajah co-authored some of the interactive books. Dorf and Lazos have no affiliation with Zyante. The teaching participants were early adopters of the interactive textbooks, which appeared in 2012/2013, which is how cross-semester analyses were enabled.

Discussion

A limitation of this analysis is the challenge of accounting for possible systemic differences between students in the compared course offerings, which may have impacted student performance. Though, the large sample size, multiple universities, and multiple types of courses mitigate this limitation.

The student performance may be due to students preferring interactive textbooks. Edgcomb[4] found that students reported higher learning-engagement with interactive content than static content, and students voluntarily spent nearly double the amount of time with the interactive content than static content.

Advice echoed by the participant instructors is that if teaching with an interactive textbook, consider assigning activities worth in total 2.5 - 10% of the overall course grade. The participant instructors reported that the points provide sufficient incentive to students to complete the activities. Also, the points for completing activities can replace traditional reading quizzes, saving lecture time and reducing grading time.

If possible, the assigned activities would be due before class. The participant instructors reported that students, who completed activities before lecture, already had a grasp on the fundamentals; so more class time was spent on higher-level discussions.

Future work includes continuing cross-semester comparisons, and examining combinations of interactive textbooks with flipped classrooms.

Conclusion

The combined analysis of 1,945 students at four introductory programming courses at three universities showed that replacing static textbooks (or static online materials) by an interactive textbook, with most other course features remaining the same, resulted in substantial improvements in exam scores, project scores, and overall letter grade, with extremely strong statistical significance. The individual cases studies showed many cases of dramatic improvements, especially for the lowest-quartile of a class, such as a nearly full letter-grade improvement for one class.

Acknowledgement

This material is based upon work supported by the National Science Foundation under SBIR Grant Nos. 1315094 and 1430537.

References

[1] Boundless. https://www.boundless.com/. February 2015.

[2] Broschat, S.L. Interactive Software for Undergraduate Electromagnetics. Education, IEEE Transactions, Volume 36, No. 1, pgs. 123 – 126, 1993.

[3] Chen, X. and M. Soldner. STEM Attrition: College Students' Paths Into and Out of STEM Fields (NCES 2014-001). National Center for Education Statistics, Institute of Education Sciences, U.S. Department of Education, Washington, DC, 2013.

[4] Edgcomb, A. and F. Vahid. Effectiveness of Online Textbooks vs. Interactive Web-Native Content. 2014 ASEE Annual Conference, 2014.

[5] Flat World Knowledge. http://catalog.flatworldknowledge.com/. September 2014.

[6] For Dummies eLearning. https://learn.dummies.com/. February 2015.

[7] Freeman, S., S.L. Eddy, M. McDonough, M.K. Smith, N. Okoroafor, H. Jordt, and M.P. Wenderoth. Active Learning Increases Student Performance in Science, Engineering, and Mathematics. Proceedings of the National Academy of Sciences, vol. 111, no. 23, pgs. 8410 – 8415, 2014.

[8] Galles, D. Data Structure Visualizations. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html. February 2015.

[9] Hagerty, G. and S. Smith. Using the Web-based Software ALEKS to Enhance College Algebra. Mathematics and Computer Education, Volume 39, No. 3, pgs. 183 – 194, 2005.

[10] Inkling. https://www.inkling.com/. February 2015.

[11] Inside Higher Ed. Options Don't Stem Textbook Woes. https://www.insidehighered.com/news/2014/01/28/textbook-prices-still-crippling-students-report-says. Written January 2014, accessed February 2015.

[12] Jarc, D.J. http://nova.umuc.edu/~jarc/idsv/lesson1.html. February 2015.

[13] Kloss, J. http://www.cs.jhu.edu/~goodrich/dsa/trees/btree.html. February 2015.

[14] Learning with Python: Interactive Edition 2.0. http://interactivepython.org/courselib/static/thinkcspy/index.html. February 2015.

[15] Mukundan, R. http://www.cosc.canterbury.ac.nz/mukundan/dsal/GraphAppl.html. February 2015.

[16] Porter, L., C. Bailey-Lee, and B. Simon. Halving Fail Rates using Peer Instruction: A Study of Four Computer Science Courses. Proceeding of the 44th ACM technical symposium on Computer science education. ACM, 2013.

[17] OpenDSA: Open Source Interactive Data Structures and Algorithms. http://algoviz.org/OpenDSA/. February 2015.

[18] Open Learning Initiative at Carnegie Mellon University. http://oli.cmu.edu/. February 2015.

[19] Senack, E. Fixing the Broken Textbook Market. U.S. Public Interest Research Group, Student PIRG, January 2014.

[20] Simon, B., S. Esper, L. Porter, and Q. Cutts. Student Experience in a Student-Centered Peer Instruction Classroom. Proceedings of the ninth annual international ACM conference on International computing education research. ACM, 2013.

[21] Simon, B., M. Kohanfars, J. Lee, K. Tamayo, and Q. Cutts. Experience report: peer instruction in introductory computing. Proceedings of the 41st ACM technical symposium on Computer science education. ACM, 2010.

[22] Simon, B., J. Parris, and J. Spacco. How we teach impacts student learning: peer instruction vs. lecture in CS0. Proceeding of the 44th ACM technical symposium on Computer science education. ACM, 2013.

[23] VisuAlgo from National University of Singapore. http://www.comp.nus.edu.sg/~stevenha/visualization/. February 2015.

[24] Weil, E. How Students Really Buy Textbooks. The New York Times. Published 2010. Accessed 2014. http://www.nytimes.com/roomfordebate/2010/7/25/the-real-cost-of-college-textbooks/how-students-really-buy-textbooks

[25] WileyPLUS. https://www.wileyplus.com/. February 2015.

[26] Wood, S.L. A New Approach to Interactive Tutorial Software for Engineering Education. Education, IEEE Transactions, Volume 39, No. 3, pgs. 399 – 408, 1996.

[27] Woody, W.D., D.B. Daniel, and C.A. Baker. E-books or textbooks: Students prefer textbooks. Journal of Computers and Education 55, no. 3, pgs. 945 – 948, 2010.

[28] zyBooks. https://zybooks.zyante.com/. February 2015.