Evaluating Online Tutorials for Data Structures and Algorithms Courses

Dr. Simin Hall, Virginia Tech

Dr. Simin Hall is a Research Assistant professor in the Department of Mechanical Engineering (ME) at Virginia Tech (VT). Currently she is collaborating with Dr. Cliff Shaffer in Computer Science Department on a National Science Foundation funded TUES project to improve instruction in Data Structures and Algorithms (DSA) courses. Her applied research in education is in cognitive functioning using online learning technologies. She has redesigned two undergraduate courses in Thermodynamics for online/Distance delivery at the ME Department at VT. In 2010, with an education grant from Nuclear Regulatory Commission (NRC) she completed the online design of the Graduate nuclear engineering certificate program. In 2011, the new education grant from NRC, allowed initiating the design of two new nuclear graduate courses for the Master program. She maintains research and publishing tracks in nascent interdisciplinary trust concepts, eLearning, and innovative teaching, learning in fields of statistics and research methods, engineering, medical fields, and assessment methods.

Prof. Clifford A. Shaffer, Virginia Tech

Dr. Shaffer received his PhD in Computer Science from University of Maryland, College Park in 1986. He is currently Professor of Computer Science at Virginia Tech, where he has been since 1987. He directs the AlgoViz and OpenDSA projects, whose goals respectively are to support the use of algorithm visualization in the classroom, and the development of a complete online collection of interactive tutorials for data structures and algorithms courses. His research interests are in Computational Biology and Bioinformatics, Problem Solving Environments, Digital Education, Algorithm Visualization, Hierarchical Data Structures, Algorithm Design and Analysis, and Data Structures.

Mr. Eric Fouh, Virginia Tech

PhD student in Computer Science at Virginia Tech

Mai Hassan ElShehaly, Virginia Tech

PhD candidate Computer Science Department

Daniel Breakiron, Virginia Tech

Evaluating Online Tutorials for Data Structures and Algorithms Courses

Abstract

We present results from a study investigating the role of online tutorials for data structures and algorithms (DSA) courses in Computer Science. We used principles drawn from research and theories in disciplines such as cognitive science, motivation, and education to design the tutorials. They were developed as part of the OpenDSA eTextbook project (http://algoviz.org/OpenDSA), an open source, online system combining textbook-quality content with algorithm visualizations and interactive exercises. DSA courses emphasize dynamic processes such as how various algorithms work. OpenDSA supports presenting such content in a highly visual manner through the frequent use of slideshows, simulations, and visualizations. Students were also provided a continuous stream of automated assessment questions and interactive exercises, thus providing immediate feedback to the students on their progress.

A pilot study was conducted with students in a Computer Science course at Virginia Tech during Fall 2012. We tested three weeks of content on sorting and hashing in a quasi-experimental setting and collected quantitative and qualitative data. The data consisted of students' performance as measured by their grades, students' perceptions and opinions obtained on surveys, field notes from observing the classes, interview data at the end of the course, and the interaction logs that our system records. After the pilot test, students' average grade in the treatment group was slightly (but not significantly) better than the control group on the post test. Students' survey and interview data indicated positive feedback about OpenDSA, with the average response on how well they liked using the OpenDSA materials increasing after use as compared to a similar pre-test question about whether they would like to use such materials.

1. Introduction

We present our first evaluation results from an ongoing research and development project that aims to fundamentally improve instruction in Data Structures and Algorithms (DSA) courses. DSA topics play a central role in every Computer Science curricula[1], defining the transition from learning programming to learning computer science. However, students often find this material difficult to comprehend because so much of the content is about the dynamic process of algorithms, their effects over time on data structures, and their analysis (determining their growth rate with respect to input size). Dynamic process is difficult to convey using static presentation media such as text and images in a textbook. During lecture instructors typically draw on the board, trying to illustrate dynamic process through words and constant changes to the diagrams. Many students have a hard time understanding these explanations at a detailed level. Another difficulty is the lack of practice with problems and exercises that is typical with DSA courses. Since the best types of problems for such courses are traditionally hard to grade, students normally experience only a small number of homework and test problems, whose results come only long after the student gives an answer. The dearth of feedback to students regarding whether they understand the material compounds the difficulty of teaching and learning DSA.

There is a long tradition among CS instructors of using computer software in the form of Algorithm Visualization (AV) [2-4] to convey dynamic concepts. AV has been used as a means both to deliver the necessary dynamic exposition, and to increase student interaction with the material[5]. However, despite the fact that surveys show that instructors are positive about using AVs in theory, and students are overwhelmingly in favor of using AVs when given the opportunity, many advocates have been disappointed at the relatively low level of sustained adoption for AV in actual classes. Surveys of instructors[2, 6] show many impediments to AV adoption, including finding and using good materials, and "fitting them" into existing classes. The OpenDSA project[7,8] seeks to remedy these impediments by providing complete units of instruction on all traditional DSA topics, available as a series of tutorials implemented using HTML5 technology. In this way, the modules are available for use on any modern browser with no additional plugins or software needed, and can even run on tablets and many mobile devices. OpenDSA modules combine content materials in the form of text, slide show, simulation, and various types of assessment questions, thus, we posit they offer solutions to all of the problems discussed above. They provide easy access, they present dynamic process with dynamic visualization, and they allow for far more practice with the material than has traditionally been possible.

From a teaching perspective, OpenDSA tutorials are designed to deliver course content incrementally, and balance teaching the dynamic processes with interactive assessment activities that provides immediate feedback to students. From a learning perspective, they are intended to move students from a passive stance in a lecture-type classroom setting to an active position of constructing learning and tracking their own comprehension through immediate feedback received from the exercises.

The theory of change driving the design and implementation of these tutorials is to encourage students' engagement with the content materials and involve them in the loop of assessment as active participants in such a way that they as well as their instructors know that they are learning. According to Bandura's[9] cognitive theory of self efficacy, instruction that allows students to check their own progress at a designated level of proficiency impacts motivation positively. Constructivist theory[10, 11] suggests that the exchange of timely feedback can encourage students to modify their work. Lovett and Greenhouse[12] show that receiving feedback and comments on the steps of learning have significant influence on learning compared to only receiving feedback from the instructor on the performance.

In the next section, details of the tutorial modules are discussed. The methods section presents a wide variety of results from testing these tutorials in a quasi-experimental setting during Fall 2012. The discussion section provides our plan for future testing and further development of OpenDSA.

2. Tutorial Modules

Tutorial modules tested in this study represent a tight integration of content, interactive exercises, and assessment. This presents an innovative way to improve the state of the art in AV and teaching of DSA, as no project previously has tried to integrate AVs with text and assessment to this degree. A module roughly corresponds to a section (a couple of pages) in a traditional textbook. A typical semester course might include over 100 modules. The vision for the completed electronic textbook is for modules and their prerequisite structure to define a directed graph that makes up a Knowledge Map. It will characterize a viable self-contained electronic textbook. Instructors might use the material in a number of ways, and determining the most effective classroom approaches is part of our ongoing studies. The modules support self-study of the material, or an instructor can use them within an instructional framework of controlled pacing and feedback whether in class, through labs, or outside of class.

Each module integrates content, visualizations, and exercises and has four major components:

- Text and images to convey normal exposition.
- Presentation of dynamic process through “slideshows". The modules in this study typically contain small slideshows interspersed with text to illustrate specific aspects of the algorithm, while a summary “slideshow" at the end of the exposition section allows the student to step through a complete dynamic presentation of the algorithm.
- Significant interactive activities. The modules in this study contain two types of larger activities: a “proficiency exercise" where the student has to demonstrate proficiency in the workings of the algorithm by essentially driving a simulation of the algorithm; and a performance analysis activity where a student selects parameters to the algorithm with the goal of improving its efficiency. These are scored. Doing something wrong on a proficiency exercise means that students lose the point associated with that step, and are then shown what they should have done, so that they can repeat and progress.
- Exercises. The current modules include three types of smaller exercises. There are many of each type.

  1) Multiple Choice/TF/type-a-number questions. These typically appear in batteries of questions, and students have to get between 4 and 10 points to receive credit for that battery (depending on how many questions there are). If a student gets a question wrong,

they lose a point. So guessing is not productive. Some students caught on to the fact that in the current implementation, if they get a question that they do not like, they can refresh the browser page to get around it.

2) "Mini" proficiency exercises, consisting of a series of randomly generated problem instances. Each problem instance is like a question in a battery, in that they have to get typically 4 or 5 of them right. Again, getting one wrong deducts a point.

3) Various odds-and-ends activities that generally require accomplishing some goal, like picking values that cause something to happen in a calculation.

The pacing for all four components is controlled by the users; however, students have to finish the exercises to receive credit. Prior experience shows that when possible to gain credit in that way, some students will abuse the ability to repeat exercises and attempt to complete them by guessing[13]. All of our exercise types are designed to lead students to reflecting on their errors prior to resubmission because they lose points for wrong responses. This makes guessing ineffective. Our proficiency exercises are modeled after the TRAKLA system[5, 14] and implemented using the JSAV algorithm visualization library[15].

3. Research Methods

We viewed this study as the first step in assessing our overall approach. Our goal was to make as much progress as possible on answering a number of key questions.

- Can students learn as well or better using interactive tutorials instead of traditional lecture and textbook?
- Will students be accepting of a class focused on interactive tutorials rather than traditional lecture and textbook?
- Will our client/server infrastructure adequately support classroom use?
- What feedback do students have about the best approaches for using such interactive tutorials in courses?

A. Methodology

We employed a mixed-method research design[16] for this study. During Fall 2012 we conducted an evaluation of the use of OpenDSA in class. We compared the impact of the self-paced tutorials on learning data structure and algorithms in a quasi-experimental setting using a control and a treatment group. The control group received standard lecture and textbook for three weeks on the topics of sorting and hashing, identical to what has been typically done in this course for many years. The treatment section spent their class time working through equivalent content in the form of online material. During some class periods, the treatment group received lecture or group discussion rather than solely working on the modules. For the treatment group, doing the modules constituted a “homework” grade worth 5% of the total class score. This replaced the homework assignment normally given for this material that would have consisted of a set of exercises from the textbook. A test was administered after the three-week intervention to both sections, allowing us to compare the results of tutorials and interactive sessions in class against standard lecture condition. We compared the mean test grades for the two groups. We also administered opinion surveys (both before and after the treatment) to both sections, to examine students' perception and opinions about these tutorials as opposed to a traditional lecture environment.

This effort follows up on previous studies we have conducted that demonstrated the effectiveness of tutorials that added algorithm visualizations alone to textbook materials. In those studies, students who used the online tutorials scored significantly better on a midterm exam than students who were given traditional lecture and textbook. We anticipated that the addition of many interactive assessment exercises would provide a greater improvement than algorithm visualizations alone. In addition, the treatment group’s use of tutorials was logged to analyze the impact of OpenDSA modules on their performance and learning. These logs also allowed for improving the development of these tutorials. Three students in the treatment group also volunteered for an extensive interview with the project team members.

B. Population and Data Collection

Appropriate Institutional Review Board permission was obtained prior to collecting and analyzing the data reported in this paper.

The undergraduate population taking part in this study comprised 55 and 57 students in the control and treatment groups, respectively. In-class surveys were administered to students in both control and treatment groups before the experiment began, and another set of surveys were administered to both sections after the test (the post-test surveys were different for the two groups). Both sections received three weeks worth of instruction on sorting and hashing, and then both sections took the identical test.

The pre-survey from both groups measured a) students’ experience with online tools, content, and assessment, b) students’ perceptions of their learning in a face-to-face course compared with online/Web-based instruction, c) students’ experience with using technology or e-textbook as it relates to accomplishing course work, and d) students’ preference for lecture courses versus courses given in a lab setting.

The post survey from the treatment group measured students’ perception, enjoyment, and satisfaction with the OpenDSA modules, as well as their preference for doing the modules in class or outside the class. They also ranked the relative impact of various aspects of the course (lecture, OpenDSA, homework, tests, forums, instructor, etc.) on their learning.

Observations of the treatment group and instructor during class also served as a method of assessment. The main question serving as the basis for the observations was: how do the tutorials facilitate interactions between students, the content, and the instructor?

Data from interaction logs for the treatment group were analyzed to examine the variability in how students accessed different components of the tutorials. We attempted to establish correlations between scores on homework in OpenDSA and time spent doing the exercises.

The interview questions consisted of students' general opinion about their experience with the tutorials, types of exercises in OpenDSA, their study process, and the amount of time spent studying.

C. Results from the Midterm

For the Control group the mean and the median of the test was 70, while for the treatment group the mean was 75 and the median was 79. Statistically the difference is not significant. However, practically it replicates the previous results on similar studies in 2008 and 2009, which did show statistically significant results. In 2008-2009, students in separate sections of the same course were taught about hashing. One section was given standard lecture and textbook for one week, similar to what had been done in previous offerings of the course. The other section spent the class time working through an online tutorial combined with AVs to present the same material. The tutorial used text content taken from the course textbook, so that it was an exact match to the material being presented in the control section. However, the online tutorial heavily supplemented this text with AVs. In each of the trials, the two sections were given a quiz on hashing at the conclusion of the week of instruction. The results were positive: significant differences in performance were obtained in favor of those who used the online tutorial versus standard lecture and textbook. In the present study, students in the treatment group participated in the study for three weeks, had more content to cover, answered additional and varied assessment questions with immediate feedback, and interacted with the system more actively. The test in the 2008-2009 study focused primarily on procedural questions about hashing, while the 2012 exam tested for topics on sorting and hashing and was not limited to procedural questions. There were also greater differences in the details of how the content was presented than was the case in the 2008-2009 studies. We hypothesize that all of these differences lead to greater variance in test scores.

D. Results from the Pre-Survey

Both control and treatment sections were given nearly identical surveys regarding their past experiences with online courseware and their perceptions regarding use of interactive tutorials as we envision. We found that nearly all students have already had significant experience with online courseware, typically including online graded work using automated assessment. This is not surprising as our students are nearly all enrolled in Virginia Tech's College of Engineering, and thus have had significant exposure to math courses that are taught largely online or in a computer lab setting. While many students expressed dislike for certain online courseware practices that they have experienced, overall they appear to have a positive attitude about the potential for an online course based on interactive tutorials and exercises. On the specific question "Would you like to take a course built around such an electronic textbook?" with a scale of 1 = "No, not at all" and 5 = "Yes, very much", the mean score was 3.79 for the control section and 3.88 for the treatment section.

The treatment section also had an additional question regarding their preference for paper versus electronic textbooks. Their course textbook was available to them for free in the form of a PDF file, or could be purchased at a price of about $20 from a vendor such as Amazon. Seven of 37 respondents for this question indicated that they had available the paper version of the textbook.

E. Results from the Post-Survey

The purpose of the post-survey was to examine students' attitudes toward (1) learning content interactively using tutorial modules versus a lecture setting, and (2) the specific approaches to how the tutorials could be deployed. The treatment group was given the question "Considering your experience with the OpenDSA materials as compared to the alternative of standard lecture and textbook, please rate your preference for how the material should be presented." On a scale of 1 = "I definitely would have preferred lecture and textbook" to 5 = "I very much preferred using OpenDSA", the mean score was 4.3. We feel that this is an important result since it indicates that students who actually used the materials greatly increased their support for their experience over their anticipated preference for online tutorials as indicated in the pre-survey (where the equivalent question scored 3.88).

Asked what percentage of class time should be spent lecturing and what percentage should be spent working OpenDSA modules, the mean value aggregated over the class was 2/3 of the class time is spent on lecturing and 1/3 of the time spent on working OpenDSA modules. Students showed strong support for a class structure that involved lecture on the material while doing homework using the OpenDSA modules: Mean=1.73 on a Likert scale of 1 = "very much like it" to 5 = "very much against it". Students showed slight preference for doing the OpenDSA assignments after class: mean=3.55 on a Likert scale of 1 = "very much prefer prior to class" to 5 = "very much prefer after the class". Students gave OpenDSA the highest rank for importance toward learning gains during the semester among all features listed, followed by lecture, then projects, course notes, and then the textbook. Treatment students indicated that they spent less time preparing for the midterm than did the control group. On the open-ended question, students indicated strong positive responses about OpenDSA. As a typical example, one student wrote that "The OpenDSA textbook was very good. By far best part was the visualization exercises which did a great job helping me understand how each algorithm and sort worked." We note that these supportive responses come despite the fact that there were some technical difficulties encountered with using OpenDSA during the first week of the experiment.

F. Summary of Classroom Observations

We observed the treatment group to find out how the tutorials affect interactions between students, the content, and the instructor. On the first day, there were technical difficulties with the server, making the system hard to log into and use. Thus, there were many questions about procedural items such as logging in. These problems were resolved after the first day. However, generally, the same groups of students were interacting on the first and second days. While most students worked independently, the majority of students who did interact with other students did so in groups of two. About 1/3 of the class worked collaboratively during the in-class sessions devoted to using the tutorials.

During the first week, the treatment group exclusively used class time to work on the modules. During the second class, a consensus emerged from the students that, while they liked the tutorials, they wanted class time to include a lecture component. Thus, during the second week the class time was largely devoted to reviewing certain aspects of the content covered in the modules. The focus of the in-class discussion was on more abstract topics such as algorithm analysis and the relationships between the various algorithms being covered. This is in contrast to prior years with traditional lecture where the bulk of lecture presentation needed to cover the mechanics of how the algorithms worked. In contrast, those mechanical aspects are well covered in the tutorials and so did not need to be covered by the instructor in class. The third week of class covered the second topic, hashing. The in-class time was a mix of working the tutorials and lecture.

G. Results from the Log of data: Treatment Group

Interaction logs were analyzed to examine the variability in students' behavior when accessing the tutorials. For example, the logs can tell us how students spread their work over time (the simplest analysis is to count the number of days on which a student worked). Each exercise has a score threshold that grants "proficiency", and a module grants "proficiency" when all contained exercises have proficient status. For the homework assignment, students are given points for each slideshow viewed, each battery of multiple choice questions, each proficiency exercise, each series of "mini" proficiency exercises, and each interactive "calculator" activity. While the total assignment was worth 50 points, it is possible to receive a score of 61 if all exercises are completed correctly. Actual credit given to students was capped at 55. The reasoning was to give some extra credit for doing additional work on the one hand, but to avoid ill feeling if certain students encountered technical difficulties that prevented them from doing one or two of the exercises on the other hand. Of the 57 students who worked the modules, 48 received the full 55 points possible, and only 8 allowed themselves to receive less than 50 points.

We found from interviews and incidental discussions that students keyed to the fact that they received assurance when they finished covering the contents and exercises for a given module. A moderate correlation of 0.38 was obtained between number of days using OpenDSA and total score on the midterm. However, the correlations between other variables compared were not significant. In particular, there was a correlation of only 0.12 between total number of exercise attempts and the midterm score, and 0.25 between number of correct exercise attempts and the midterm score.

We hope to perform further analysis on the log data to find more sophisticated relationships between behavior and performance. We can determine from the logs when students receive their assignment points, and can then determine if they do the bulk of the work at the last minute. We can also measure the amount of time spent on exercises or number of exercises done after proficiency for the exercise has been granted. These activities can be classified as "studying" rather than work done for assignment credit. We can measure the number of successful vs. unsuccessful exercise attempts. We are still developing the tools that will let us perform these analyses. We are also working to develop visual tools that track student performance over time. Eventually, we hope to be able to analyze more abstract behavior. For example, it is possible for students to bypass the text and go directly to the assessment questions. Recognizing this and measuring it will require careful analysis of relative time of user interaction events.

H. Results from the Interviews

Interviews were conducted with three students selected from the treatment group. One subject routinely procrastinated on all coursework and did the OpenDSA assignment primarily on the last day. One subject had expressed definite opinions about using the modules in class during the course of the experiment. One subject had previously interacted with us a lot to report bugs and usability issues. Thus, we hoped that these interviews would represent a range of student opinions.

We asked the subjects about their general opinion of working with OpenDSA. All expressed positive views of the materials. In one way or another, they all indicated that the modules "fit their learning style", though they probably mean different things when they say that. The biggest complaint was that the number of multiple choice questions needed to be expanded, so that students could not just memorize the answers. Another common complaint was that OpenDSA modules had not been available for content covered after the end of the three week experiment. In general, while some initial problems with using the system were noted, the subjects did not see those as important issues since they were cleared up during the course of the experiment.

One subject commented that he was a visual learner and normally preferred course notes to textbook. He indicated that he used the modules mostly during the class, until near the time when he had to complete the modules for homework credit. However, for test preparation he found the modules and the exercises very helpful. The proficiency exercises resembled games and he liked the multiple choice questions as confirmation of his knowledge. He indicated that it would be nice to add a feature where he could send questions to the instructor.

One subject was against using the modules for in-class work, preferring that they be done at home. She indicated that she preferred to study in a quiet environment. None of the subjects indicated support for group work with the modules. All subjects indicated they had better retention of the content from using OpenDSA versus what they believe they would have had from traditional textbook and lecture.

Interview feedback, surveys, and log data are all consistent that the amount of total time and effort to complete the online materials is roughly on par with that required when doing a traditional lecture-based class with written homework, and the amount of time spent was considered appropriate by the students. Little support was expressed during interviews or on surveys in favor of additional features for note taking, annotating, or social interaction support such as sending comments to classmates about the material.

4. Discussion

The results of our study confirmed that online tutorials can be an effective method of learning, as demonstrated by test scores. This is not a new finding. What is important about our results is the feedback received from students, and the emerging picture about student use of tutorials that we can deduce from the log data.

Students strongly endorse the use of AVs mixed with text, especially when used in an environment when they can process the material at their own rate. They especially appreciate AVs' ability to present procedural aspects of algorithms, and OpenDSA's exercises that allow greatly increased practice as well as interaction with the material. They also appreciate the immediate feedback that indicates to them how secure they are in their knowledge of the material.

One major consideration that might influence the results from using OpenDSA is the class structure. In our case, how much impact does coming to class and doing the tutorial in "lab" setting have on the results? The outcome could be quite different for a student just reading the material and working through the visualizations on their own, where their self-discipline might well not be sufficient to make them provide the necessary amount of time and attention. Likewise, the controlled environment of attending lecture before reading the textbook on one's own is likely to have a major difference compared to just reading the book on one's own. Our results confirm Naps & Grissom' results[17] that students' learning is enhanced with extended exposure to a tool rather than an isolated experience. Their findings showed that for AVs to be effective, they must be integrated into a course.

We were rather surprised to learn of students' strong desire to continue with some level of lecture in the course, even while they were enthusiastic about using the OpenDSA tutorials. There is a wide range of opinions on this, with some students preferring just doing tutorials at home. But the bulk of students want some lecture, or even all classes to be lecture with all tutorial work done at home. Aggregating survey responses across the entire class, the average was 1/3 class use of tutorials and 2/3 lecture. Based on this feedback, we are modifying our plans for Fall

2013, where we hope to conduct the entire semester class with the aid of OpenDSA. Instead of creating a largely lab-based class as we originally envisioned, we are now planning to teach the class using largely lecture, but directly tied to daily use of OpenDSA modules. Our feedback is fairly split regarding whether the day's module assignment should be due before the content is presented (as "prior reading") or after the class (as summary and consolidation). We will experiment with both approaches.

Logging students' interactions and tying tutorial progress to their grades could better motivate students to increase their participation. "Students typically do better when they know their work is being evaluated"[17]. Some of the students' responses to the usefulness of tutorials were inconclusive. This has been observed in Naps & Grissom'[17] study that student's perception of learning may not be useful in predicting the true effectiveness of a tool. Log data will help us to relate actual student behavior and performance. We will better be able to design the tutorials to encourage students to read all of the content, and we will be able to recognize which exercises are giving students unexpected problems (perhaps due to poor user interface design).

5. Summary

Pedagogical assessment when deploying new courseware is crucial to success. The classical approach to design and delivery of courses relegates assessment of teaching/learning objectives to the end. We have used a parallel model of development by integrating the assessment design into the iterative phases of the design of both the tutorial content and the exercises. We envision this study as only the first major round of formative evaluation for the system. This study provides a systematic approach to gaining information that will allow us to integrate our course materials with course objectives and teaching strategies. These results have already yielded the following benefits to us:

- Affirmed that the approach of online tutorials will be well received by our students.
- Made us reassess our planned teaching strategy for the next deployment, to include more in-class lecture on the more difficult concepts so as to reinforce the online materials.
- Gave us insight on how to improve our server-side logging and scoring infrastructure.
- Gave us insight on what log data to collect for our next round of testing, and what tools we need to analyze it.
- Allowed us to perform fine-grained tuning of the specific exercises used in terms of things like making sure that students take the right amount of time to complete and adequately cover the material.

Our current plan is for additional small-scale testing in Spring and Summer 2013, followed by a full semester's worth of online materials to be deployed in Fall 2013. We hope to introduce navigation between modules using a Knowledge Map rather than traditional hierarchical table of contents. This approach has been shown[18] to help students activate prior knowledge and integrate that with new materials. Seeing relationships between the pieces of knowledge enforces better performance and supports deeper learning. We are in the process of arranging deployment of our materials at a number of other campuses.

Acknowledgment: This study was supported by NSF/TUES grant DUE-1139861. The authors also would like to acknowledge the Institute of Distance & Distributed Learning, the Center for Instructional Development and Educational Research, and the Institute for Society, Culture, and Environment at Virginia Tech for funding and support for this research. All opinions expressed are those of the authors and not necessarily those of the National Science Foundation or the other funding agencies supporting this research.

Bibliography

1. CS2008 Review Taskforce (2008). Computer science curriculum 2008: An interim revision of CS 2001. http://www.acm.org/education/curricula/ComputerScience2008.pdf.

2. T.L. Naps, G. Rössling, and nine more authors (2002). Exploring the role of visualization and engagement in computer science education. Iin *ITiCSE-WGR '02: Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education*, pages 131-152.

3. C.A. Shaffer, M.L. Cooper, A.J.D. Alon, M. Akbar, M. Stewart, S. Ponce, and S.H. Edwards. (2010). Algorithm visualization: The state of the Field. *ACM Transactions on Computing Education*, 10:1-22.

4. E. Fouh, M. Akbar, and C.A. Shaffer (2012). The Role of Visualization in Computer Science Education, *Computers in the Schools 29,* Issue 1-2, 95—117.

5. L. Malmi, V. Karavirta, A. Korhonen, J. Nikander, O. Seppala, and P. Silvasti (2004). Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. *Informatics in Education*, 3(2):267- 288.

6. C.A. Shaffer, M. Akbar, A.J.D. Alon, M. Stewart, and S.H. Edwards (2011). Getting Algorithm Visualizations into the Classroom in *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE'11)*, Dallas TX, March 9, 129—134.

7. C.A. Shaffer, V. Karavirta, A. Korhonen and T.L. Naps (2011). OpenDSA: Beginning a Community Hypertextbook Project, in *Proceedings of 11th Koli Calling International Conference on Computing Education Research*, Koli National Park, Finland, November 17-20, 112—117.

8. C.A. Shaffer, T.L. Naps, and E. Fouh (2011), Truly Interactive Textbooks for Computer Science Education in *Proceedings of the Sixth Program Visualization Workshop*, Darmstadt, Germany, June 30, 97—103.

9. A. Bandura (1986). *Social Foundations of thought and action: A social cognitive theory.* Englewood Cliffs, NJ: Prentice Hall.

10. B. Rogoff (1991). Social interaction as apprenticeship in thinking: guidance and participation in spatial planning, in *Perspectives on socially shared cognition*, L.B. Resnick, J.M. Levine, and S.D. Teasley, Eds. Washington, DC: APA, 349-383.

11. K. Swan (2005). A Constructivist model for thinking about learning online, in *Elements of Quality Online Education: Engaging Communities*, Volume 6 in the Sloan-C Series Sloan-C Foundation, pp. 13-31.

12. M.C. Lovett and J.B. Greenhouse (2000), Applying cognitive theory to statistics instruction, *The American Statistician*, *54*(3), 196-211.

13. V. Karavirta, A. Korhonen, and L. Malmi (2005), Different learners need different resubmission policies in automatic assessment systems, in *Proceedings of the 5th Annual Finnish / Baltic Sea Conference on Computer Science Education*, pages 95-102. University of Joensuu.

14. A. Korhonen, L. Malmi, P. Silvasti, J. Nikander, P. Tenhunen, P. Mrd, H. Salonen, and V. Karavirta (2003). TRAKLA2. http://www.cs.hut.fi/Research/TRAKLA2/.

15. V. Karavirta and C.A. Shaffer (2013), JSAV: The JavaScript Algorithm Visualization Library, in *Proceedings of the 18th Annual Conference on Innovation and Technology in Computer Science Education (ITICSE 2013)*, Canterbury, UK, July 2013.

16. J.W. Creswell and V.L. Plano Clark (2007), *Designing and Conducting Mixed Methods Research,* Sage Publication.

17. T.L. Naps and S. Grissom (2002). The effective use of quicksort visualizations in the classroom, *Journal of Computing Sciences in Colleges*, 18 (1), 88-96.

18. S.A. Ambrose and four more authors (2010). *How Learning Works: Seven Research-Based Principles for Smart Teaching,* The Jossey-Bass Higher and Adult Education Series.