Computers in Human Behavior xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Exploring students learning behavior with an interactive eTextbook in Computer Science courses

Eric Fouh *, Daniel A. Breakiron, Sally Hamouda, Mohammed F. Farghally, Clifford A. Shaffer

Department of Computer Science, Virginia Tech, United States

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
eTextbook
Learning behavior
Mobile learning
Computing education

ABSTRACT

We present empirical findings from using an interactive electronic textbook (eTextbook) system named OpenDSA to teach sophomore- and junior-level Computer Science courses. The web-based eTextbook infrastructure allows us to collect large amounts of data that can provide detailed information about students' study behavior. In particular we were interested in seeing if the students will attempt to manipulate the electronic resources so as to receive credit without deeply going through the materials. We found that a majority of students do not read the text. On the other hand, we found evidence that students voluntarily complete additional exercises (after obtaining credit for completion) as a study aid prior to exams. We determined that visualization use was fairly high (even when credit for their completion was not offered). Skipping to the end of slideshows was more common when credit for their completion was offered, but also occurred when it was not. We measured the level of use of mobile devices for learning by CS students. Almost all students did not associate their mobile devices with studying. The only time they accessed OpenDSA from a mobile device was for a quick look up, and never for in depth study.

1. Introduction

Recent interest in MOOCs (Massive Open Online Courses) and the popularity of practice systems like Khan Academy and Code Academy is just the latest in a trend toward increased use of interactive online course materials at all levels of education. MOOCs especially drive a need for scalable methods of assessment that do not require direct involvement of limited teaching resources. While some MOOCs attempt to address the need for assessment through crowd sourcing (especially to students in the class), automated assessment of exercises provides an excellent way to scale up assessment when it is possible. For many traditional courses, a longstanding problem is lack of sufficient practice exercises with feedback to the student. Again, automated assessment provides a way to increase the number of exercises on which students can receive feedback.

Online tutorials and interactive exercise systems provide the opportunity to automatically log massive amounts of user interaction data at fine detail, to the level of individual mouse events. This wealth of information might be used in a number of ways to improve the pedagogical value of online materials. Developers can hope to learn from interactive log data simple things like what platforms or devices are most often used by students, which exercises are taking an unreasonable amount of time, or which ones appear too easy because students never get them wrong. But careful analysis of log data can hope to deduce more complex behavior. For example, by examining the times of various interactions, we should be able to determine whether students are reading the materials before attempting the associated exercises. We can hope to tell whether students are viewing all of the parts of a given visualization, or skipping through it. We should be able to tell whether students are going back to the materials to use them to study for a test by the fact that they look at them even after receiving credit for completing them.

The OpenDSA project (Fouh et al., 2014; Shaffer, Karavirta, & Naps, 2011; Shaffer, Naps, & Fouh, 2011) provides a collection of online, open-source tutorials for Data Structures and Algorithms (DSA) courses. They combine textbook-quality text with algorithm visualizations (AV) and randomly generated instances of interactive examples and exercises to provide students with unlimited practice. OpenDSA collects log data for all user interactions occurring on an OpenDSA web page. Between Spring 2013 and Spring 2014, we collected and analyzed usage data from eight courses totaling about 700 students. We present an analysis that demonstrates how log data might be used to understand how students use online tutorial systems.

* Corresponding author at: 2000A Torgersen Hall, Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, United States. Tel.: +1 (540) 231 4354.
E-mail addresses: efouh@vt.edu (E. Fouh), breakid@vt.edu (D.A. Breakiron), sallyh84@vt.edu (S. Hamouda), mfseddik@vt.edu (M.F. Farghally), shaffer@cs.vt.edu (C.A. Shaffer).

http://dx.doi.org/10.1016/j.chb.2014.09.061
0747-5632/

2. Materials and methods

The basic functional unit for OpenDSA materials is a module, which represents a single topic or part of a typical lecture, such as a single sorting algorithm. Each module is a complete unit of instruction and typically contains AVs, interactive assessment activities with automated feedback, and textbook quality text. Modules can be grouped together into chapters, such as might be found in traditional paper books. OpenDSA content is built using HTML5 and JavaScript, making it device and browser independent. AVs are built using the JavaScript Algorithm Visualization (JSAV) library (Karavirta & Shaffer, 2013). Many OpenDSA exercises are "algorithm simulations". These require that the student manipulate a data structure to show the changes that an algorithm would make on it, such as clicking to swap elements in an array or clicking on appropriate nodes in a tree or graph. The AVs and algorithm simulation exercises are specifically designed to be manipulated either through mouse and pointer interactions or touch interactions when using touchscreen devices. We generally refer to algorithm simulation exercises, as "proficiency exercises". This type of exercise were inspired and built in collaboration with the team that created the TRAKLA2 system (Malmi et al., 2004). We make use of the Khan Academy framework (http://github.com/Khan/khan-exercises) to provide support for multiple choice, T/F, and custom interactive exercises that we call "mini-proficiency" exercises. We will refer to these collectively as "KA exercises". All exercises are automatically graded and provide feedback to the user. Students can repeat exercises as many times as they want until they get credit, or even work them again after receiving credit as a study aid.

OpenDSA has been used to teach the second semester fundamental data structures and algorithms course (CS2), and also a more advanced data structures, algorithms, and analysis course (CS3) at a total of five higher education institutions in the US, Egypt, and Finland. Different instructors have used OpenDSA in different ways, but typically they used OpenDSA exercises for graded homework, and/or used AVs from OpenDSA as a lecture aid. During Spring 2013, OpenDSA was used in three course sections: two offerings of Virginia Tech's version of CS3 (we will refer to these as C1 and C2), and one offering of a CS3 course at Alexandria University in Egypt (which we will refer to as C3). OpenDSA was used as the primary resource to teach topics related to Hashing in C3, followed by a midterm that included questions on hashing. Course C1 used OpenDSA as the primary source of material for sorting and hashing (about three weeks of material). For this section, AVs (slideshows) were not given credit. There was a single assignment to complete the exercises for the two chapters, so all exercises were due at the same time. In contrast, course C2 used another textbook for the initial presentation of material on sorting. However, OpenDSA exercises were then assigned after the lecture period on sorting was complete. C2 used OpenDSA as the primary source for Hashing. C2 students received a small amount of credit for completing slideshows. We deliberately made this distinction in slideshow credit between the two sections in order to study how that affected student learning behavior, as described below.

We relied on the logged data to infer students' behavior. The OpenDSA system records about 200 different types of events. The events can be grouped into the following categories:

- Registration and login interactions (all actions such as student registration, logging into and out of the system).
- Static Content interactions (when a student loads a module page, follows a hyperlink, or navigates to another page using the navigation menu or the table of contents).
- Interactive Activities interactions (when the student clicks to advance a slideshow, clicks within an AV, etc.).
- Assessment Activities interactions (all interactions involving loading an exercise, submitting an answer, completing a step of a proficiency exercise, etc.); and
- Gradebook interactions (when students load the gradebook page to check their score).

All events are recorded with a timestamp. To assess the relationship between the use of OpenDSA and students performance, we used both log data and performance data (students' scores in written tests) from the Fall 2013 offering of a CS3 course at VT (which we refer to as C4). OpenDSA was used as the main course material, and the instructor regularly used the visualization in the classroom as lecture aide. The students performed regular mandatory OpenDSA homework and took two midterms and one final examination. OpenDSA-based homework accounted for 20% of the course final grade.

3. Student learning behavior

We sought to measure the prevalence of several behaviors that we collectively term as "credit-seeking". We believe this is different from (and possibly in conflict with) "learning" behavior. Examples of "credit-seeking" behavior include jumping straight to exercises without reading the associated text, clicking through slideshows (such as shown in Fig. 1) as quickly as possible, skipping to the end of slideshows to get credit, and using the results from AVs to complete exercises.

3.1. Not reading

We observed two general patterns of behavior related to reading the text. Ideally, students would read most or all of the text, completing exercises as they encounter them within the module. But students often skip directly to the exercises, only reading as required to get exercise credit. Note that proficiency and KA exercises are placed "behind a button" by default, in that they require that the student clicks on an interface button to reveal the exercise.

We measured the total time that a module was open before a button was clicked to reveal an exercise. We assumed that once a student clicked the button to reveal an exercise, they attempted

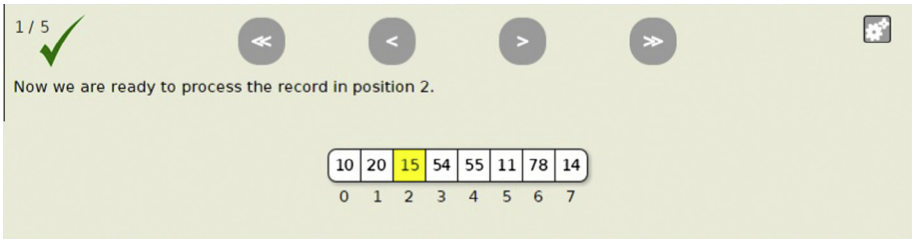

Fig. 1. Example of an OpenDSA slideshow. Standard controls allow the user to advance the slideshow by one slide, back up one slide, jump back to the beginning, or jump to the end.

Please cite this article in press as: Fouh, E., et al. Exploring students learning behavior with an interactive eTextbook in Computer Science courses. *Computers in Human Behavior* (2014), http://dx.doi.org/10.1016/j.chb.2014.09.061

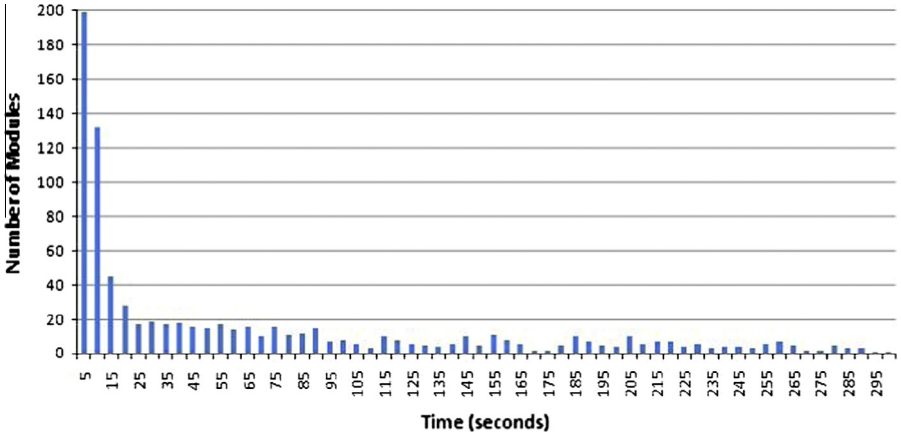

Fig. 2. Time between initial module page load and the first exercise event for C1, using 5-s bins.

Table 1
Quartiles for time spent between loading a module and starting an exercise.

Class	Quartile 1 (s)	Quartile 2 (s)	Quartile 3 (s)
C3	0–15	16–90	91–440
C1	0–10	11–70	71–290
C2	0–20	20–120	121–420

it immediately. Fig. 2 represents the distribution of all the modules with exercises attempted by C1 students, grouped by the time between the moment the page is loaded and when the students uncover the exercises. All classes showed a peak at the beginning of the histogram, indicating that a large number of students opened modules and immediately started exercises. We quantified how much the class as a whole read by identifying the number of modules in quartiles ranked by time between page load and first exercise attempt for each class. Table 1 shows that the first quartile ranges from 10 to 20 s and the second quartile from 70 to 120 s. This means that students begin exercises on 25% of modules within 20 s of loading the page and on 50% of modules within two minutes. The third quartile is roughly five to seven minutes. Note that at least some measures from the upper 25% quartile are likely to include instances where the user left the browser page open without being active. So the amount of skipping directly to exercises could be greater than the quartiles indicate. While there is naturally a high variability between the speeds at which different people read, and we lack the ability to measure passive interaction (that is, time when a student is actually reading without manipulating the webpage), we can say with reasonable confidence that students read only about half of the modules, at least on their first view. In other words, typically the first time that a student works on a module, it is for credit purposes (to get credit for completing homework) rather than for the purpose of understanding the material.

3.2. Clicking through slideshows

We also observe two patterns of behavior related to viewing slideshows. We term "learning" behavior to be when students read slide descriptions and navigate backward and forward as necessary to reexamine the content. We term "rushing" behavior to be when students click through the slides as quickly as possible in order to obtain credit, without paying attention to the material. Note that there are two forms of "receiving credit". One is that a student actually receives points (typically one tenth of a point per slideshow, with a module worth typically 2–5 total points) when some criterion is reached to define "completion". The other is that a green checkmark appears on the slideshow once the criterion is reached to define "completion". In addition, once all activities on a module page (including slideshows and exercises) have completion credit, the module shows a "module complete" message on the page, and the gradebook and table of contents also mark the module as "complete". For some students, "checking off" the module seems to be intrinsically important (separate from perhaps learning the material).

To determine whether students rushed through slideshows, we examined the mean time per slide on the occasions where a student first completed a slideshow. In this study, we did not analyze whether students who rush through slideshows return later and exhibit learning behavior after obtaining completion credit. We will analyze that aspect of student behavior in subsequent studies.

Each slide within a slideshow contains text, most often a single line. Based on the time spent we infer reading of the text or not as follows:

- If time spent on a slide is less than 8 s, we conclude that students did not read the text on the slide (no reading).
- If time spent on a slide is between 8 and 15 s, we infer that the student read the slide thoroughly (reading).
- If time spent is greater than 15 s, we conclude that the student stepped away from the computer while the slideshow was up (stepped away).

Even though we choose our threshold arbitrarily, we think it provides a good estimate for the occurrence of reading.

Fig. 3 shows the distribution of times for the C1 section. We see that more than 90% of the slide have a view time of less than 7 s. We also examined the class behavior through the use of quartiles of slideshows ranked by view time. Table 2 shows the different quartiles for C1, C2 and C3 courses.

For all three classes, 25% of slideshows were completed with a mean-time-per-slide of one second or less. The second and third quartiles for both Virginia Tech classes indicate that 50% were completed with a mean less than two seconds and 75% with a mean less than three seconds. Both the second and third quartiles for the C3 class were larger than the third quartile of the Virginia Tech classes, indicating that students in this class tended to spend longer on slideshows. We hypothesize that this occurs because these students are non-native English speakers.

Overall, it appears that a sizeable number of students in all classes quickly click through the slideshows. Calculating mean time per slide assumes constant time is spent on each slide, which is unlikely. Given the small means, this technique is unable to differentiate between students who click through all the slides quickly

Please cite this article in press as: Fouh, E., et al. Exploring students learning behavior with an interactive eTextbook in Computer Science courses. *Computers in Human Behavior* (2014), http://dx.doi.org/10.1016/j.chb.2014.09.061

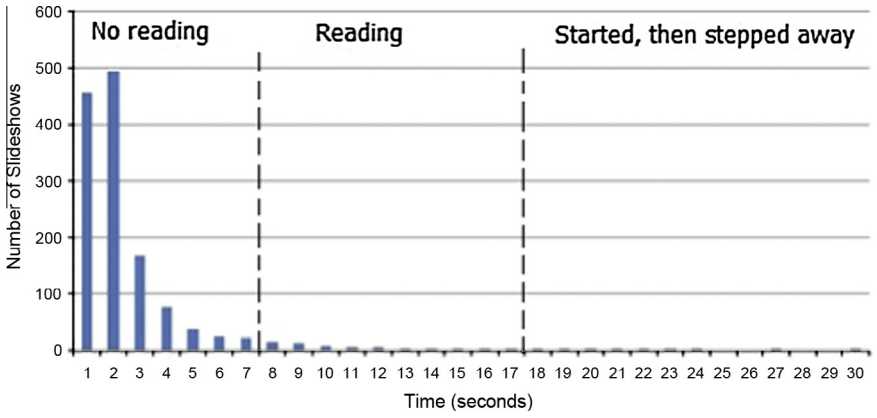

Fig. 3. Mean time per slide, C1.

Table 2
Quartiles for mean time per slide.

Class	Quartile 1 (s)	Quartile 2 (s)	Quartile 3 (s)
C3	1	4	8
C1	1	2	3
C2	1	2	3

and those who initially exhibit learning behavior and then quickly click through the rest of the slides once they grasp the necessary concepts. Additionally, distractions could cause a single slide time to be significantly longer than the others, distorting the overall mean time. Future work using more reliable event data should examine individual slide times to account for these factors.

An important strategy that we have begun incorporating into the design of our modules is to use many short slideshows that each illustrates one particular aspect of the topic, rather than a single, longer slideshow. We hypothesize that this will lead to more attention given to each slideshow.

3.3. Skipping to the end of slideshows

We knew prior to our experiment that we had a bug that allows students to obtain credit for slideshows without viewing all the slides by just jumping to end of the slideshow and clicking on the "forward" button once to get credit. We intentionally left this bug in place during our experiment in order to quantify how much students discovered and took advantage of it to artificially gain credit through this "skipping" behavior. (We have since changed completion criteria to require that the student actually click through every slide in the slideshow, which of course will change behavior from what we observed in this study.) There was a small amount of slideshow skipping in C3 and C1, and a moderate amount in C2. The key difference is that in C2, credit was given for completing a slideshow. In C3, five students (24%) skipped slideshows. While one student skipped 8 of 16 slideshows, three of the five only skipped a single slideshow. In C1, 12 of 51 (24%) students skipped slideshows. While 75% of the group skipped 5 or fewer slideshows, three students skipped 8, 15 and 28 of the 44 slideshows. The median was 3.5 slideshows skipped. In C2, 23 of 65 students (35%) skipped at least one slideshow. 31% of the C2 group skipped 5 or fewer slideshows; the class median was 8 slideshows skipped. Inspection of the event data revealed two types of behavior. Some students viewed a portion of the slideshow, possibly moving backward and forward again to examine a specific operation, and then skipped to the end. Other students skipped straight to the end without viewing any slides.

Table 3
The number of exercises attempted, completed, completed with a score above the proficiency threshold, and the number of exercises where students used an AV for assistance.

Class	Exercise	Attempts	Completions	Proficiency	Assisted
C1	Shellsort	671	67	57	1
	Mergesort	263	181	181	0
	Quicksort	1105	71	52	14
C2	Shellsort	561	90	68	2
	Mergesort	127	68	67	0
	Quicksort	1065	76	60	19

3.4. Using AVs to complete exercises

Our proficiency exercises require students to perform actions that closely resemble those demonstrated by some algorithm that is present using a related AV. We know from anecdotal data that some students ran the AVs using the exercise input, and then mimicked the AV output in order to complete the exercises. We sought to quantify this behavior. We determined that eight C1 students (16%) and seven students in C2 (11%) used AVs to assist in exercise completion. Table 3 provides details. We considered how the group of students who used AV assistance compared with their non-assisted classmates in terms of total time required for proficiency. Using each student's median time for proficiency exercises, we found the assisted group in C1 had a median of 127.5 s, and the non-assisted group had a median of 84 s. The assisted group in C2 had a median of 119 s, and the non-assisted group had a median of 77.5 s. In other words, copying the results from the AV to the exercise slowed the students down (though we cannot determine from this how long the other students spent in gaining the necessary knowledge to successfully complete the exercise).

3.5. When do students study

We analyzed data from the CS2 course offering at Virginia Tech during spring 2014 to find out when students actually study. We will refer to this course as C5. In C5, all the assignments were due Sunday at 11:59 PM, while the lecture sessions happened on Mondays and Wednesdays. The students usually had a week to complete a given assignment associated with the prior week's lectures. We analyzed our log data to see when the students usually work on OpenDSA exercises.

Figs. 4 and 5 show that most exercises were done on the due date. The graph in Fig. 6 confirms that most interactions with OpenDSA happened on Sundays. The relative surge of activity on Tuesday, May 13 corresponds to the course final examination. This

Please cite this article in press as: Fouh, E., et al. Exploring students learning behavior with an interactive eTextbook in Computer Science courses. *Computers in Human Behavior* (2014), http://dx.doi.org/10.1016/j.chb.2014.09.061

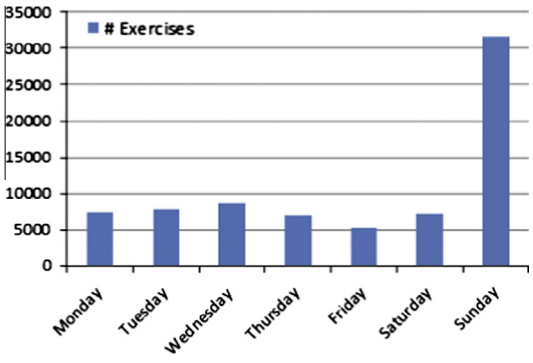

Fig. 4. Daily distribution of exercises performed.

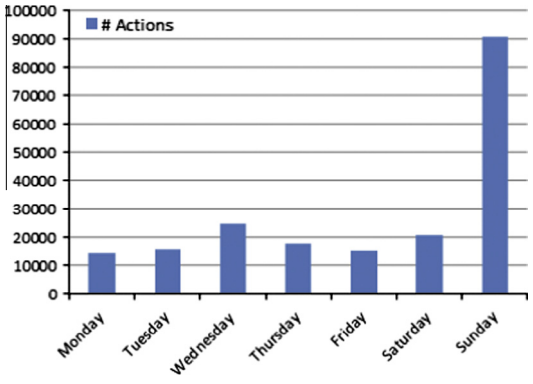

Fig. 5. Daily distribution of all interactions.

activity represents use of the materials as a study aid, rather than for completing graded assignments.

On Sundays, most of the work is done between 2 pm and midnight as shown in Fig. 7. The above results confirmed that most students waited until the last day to do their homework.

4. Student performance

In our opinion, the single most important pedagogical feature of interactive eTextbooks and OpenDSA is the interactive exercises. They provide the students with a mastery-based experience on a topic, since they allow the students to try exercises as many times as they want. We investigated the relationship between OpenDSA exercises and student performance on tests. We only focused on students' declarative knowledge, hence only students' performance on written tests are included in our analysis.

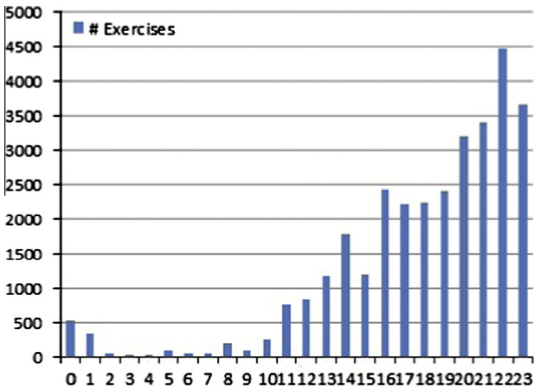

Fig. 7. Hourly distribution of exercises performed on Sundays.

OpenDSA includes three categories of exercises: Simple Questions, Proficiency exercises, and Programming exercises. Simple questions are similar to traditional T/F, multiple choice, or fill-in-the-blank questions. Proficiency exercises are visual algorithm simulation activities. We collected the following data for each student:

- Total number exercises attempted (Total.KA for simple questions, Total.KAV for KA-based proficiency exercises, and Total.PE for JSAV-based proficiency exercises).
- Total number of completed exercises (Correct.KA for simple questions, Correct.KAV for KA min proficiency exercises, and Correct.PE for JSAV-based proficiency exercises).

We collected and analyzed the data from C4. The course's OpenDSA instance had a total of 95 required exercises, of which 36 were KA simple questions, 30 were KA mini proficiency exercises, 26 were JSAV-based proficiency exercises, and 3 were other interactive activities.

We divided the data into two groups: M1 for all activity until the first midterm examination and M2 for all activity between the first and second midterms. The final exam was on sections with very little or no interactive elements in OpenDSA, so it was not included in our analysis.

Before Midterm 1, students were required to solve a total of 15 KA simple question sets, 12 KA proficiency exercises, and 6 JSAV proficiency exercises. Between Midterm 1 and Midterm 2, they had to complete 19 KA simple question sets, 18 KA mini proficiency exercises and 20 JSAV full proficiency exercises. Figs. 8 and 9 show the performance distribution. The *x*-axis of the Correct.KA, Correct.KAV, and Correct.PE histograms represent the number of exercises correctly completed. In the case of KA simple

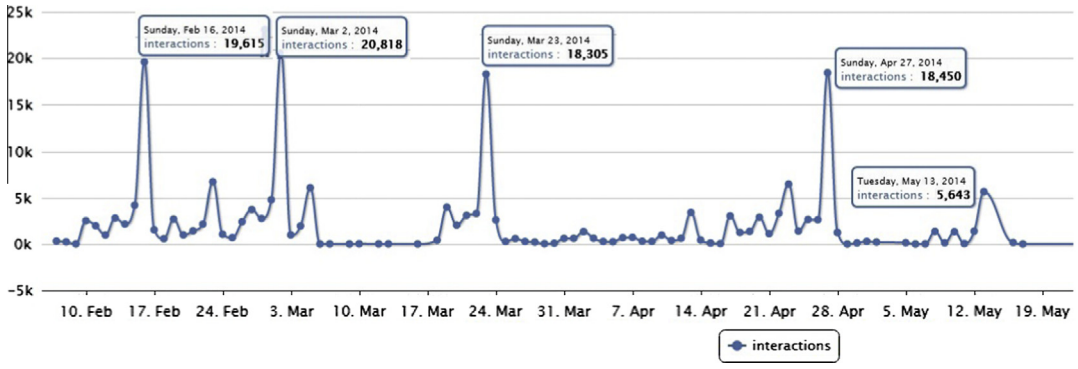

Fig. 6. C5 semester-long OpenDSA activity distribution graph.

Please cite this article in press as: Fouh, E., et al. Exploring students learning behavior with an interactive eTextbook in Computer Science courses. *Computers in Human Behavior* (2014), http://dx.doi.org/10.1016/j.chb.2014.09.061

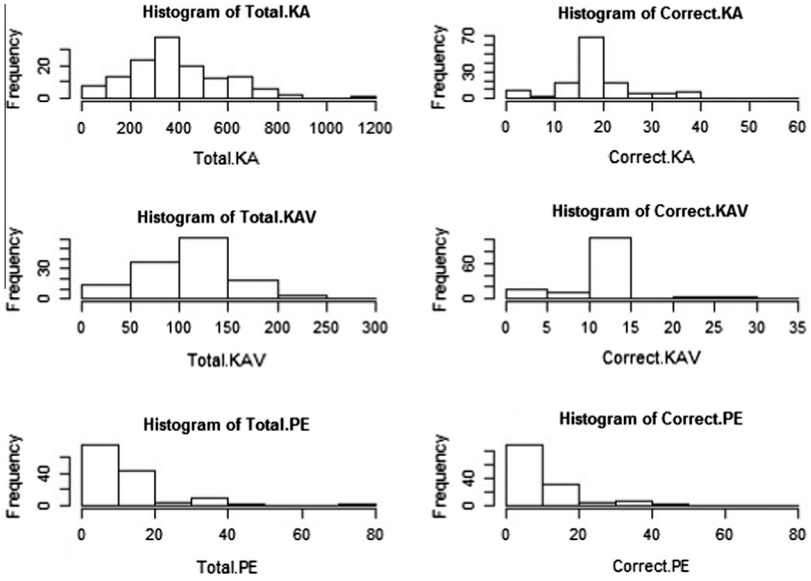

Fig. 8. C4 – Midterm 1 distribution.

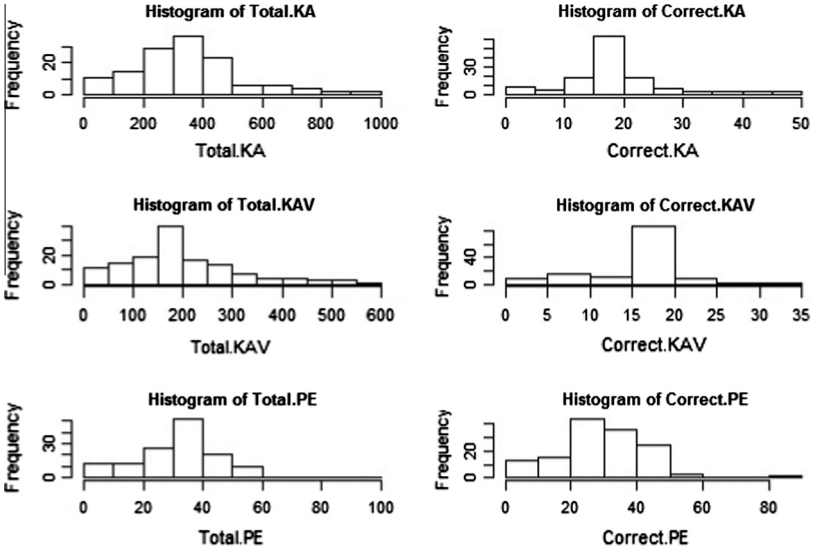

Fig. 9. C4 – Midterm 2 distribution.

questions and KA proficiency exercises it is the number of times a student reached the proficiency score. For JSAV proficiency exercises, it is the number of times a student reached the proficiency threshold (correctly simulated 90% of an algorithms step).

We see from the histograms that most students correctly completed most exercises only one time regardless of the type of question. The exception was the JSAV proficiency assignments before the second midterm, where the majority of students correctly completed exercises more than once. We can explain the exception by the fact that Midterm 2 covered the Sorting and Hashing chapters. Both included concepts that were fairly new to students. Midterm 1 covered topics where the students had had previous exposure (linear data structures, algorithm analysis and binary trees).

Since there was no penalty associated with attempting an exercise several time, we expected to see a difference between the students who correctly completed the exercises "post-proficiency".

We found a low correlation between the amount of correct exercises (regardless of the type) completed by the students and exam scores as shown in Table 4. The negative correlation between the total number of KA simple question attempted suggests that some students were just trying out all the provided answers until they find the right one, without deeply thinking about the exercise's answer, or reading the module text.

Table 4
C4 – correlation table.

Variables	Mid 1	Mid 2
Total.KA	−0.11	0
Correct.KA	0.32	0.29
Total.KAV	0.09	0.01
Correct.KAV	0.29	0.25
Total.PE	0.15	0.20
Correct.PE	0.19	0.20

We wanted to know if the analysis of student scores will show different patterns of OpenDSA exercise use. We computed the

Please cite this article in press as: Fouh, E., et al. Exploring students learning behavior with an interactive eTextbook in Computer Science courses. *Computers in Human Behavior* (2014), http://dx.doi.org/10.1016/j.chb.2014.09.061

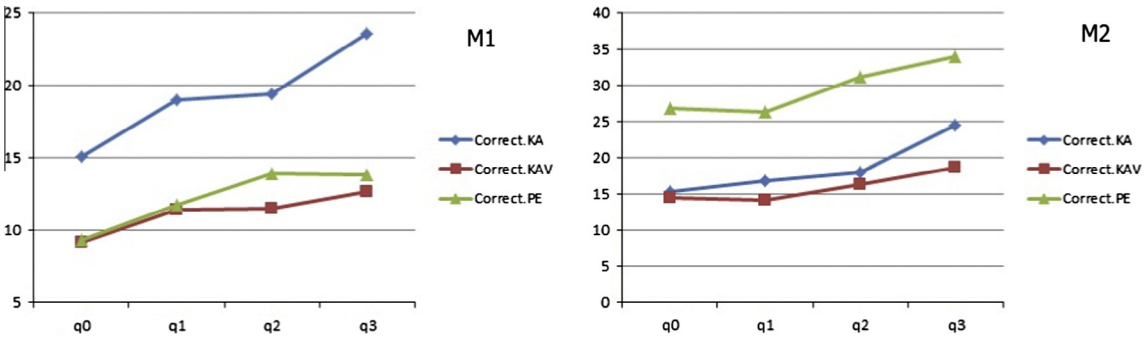

Fig. 10. C4 – M1 and M2 variable means per quartile.

average number of correct exercises performed by students grouped by midterm score quartile. As shown in Fig. 10, students with higher scores correctly completed more exercises than those with lower grade. The difference in the average number of correct exercise between quartiles was significant at the $p < 0.05$ level for KA simple questions and KA mini proficiency exercises. However, the difference was not statistically significant for JSAV proficiency exercises.

We can say that when used as the main course material, a high number of correct exercises (above the minimum level required to receive homework credit) is associated with higher grades on written tests. That is, using the exercises for additional study appeared to help students on the midterm. The above findings can be used to detect struggling students, since they are less likely to redo an exercise. In addition, students with the lowest grades have the lowest "number of correct exercises to number of exercises attempted" ratio.

5. Students use of mobile devices

As handheld wireless mobile computer become more ubiquitous, their potential use and benefit in education is being examined. Learning technology developers and users are excited about the opportunities provided by mobile devices (Roschelle, 2003). In their meta-analysis of mobile learning study, (Wu et al., 2012) concluded that most research in mobile learning studies resulted in positive outcomes. Also, they found that mobile learning was most frequently used by students in the professional and applied sciences field. However, Computer Science was the second most investigated discipline (after languages and linguistic) in regard to mobile learning. Other studies show that college students want to use mobile devices more for learning and hope that their instructors and institution will create an environment that promotes the use of mobile devices (Dahlstrom, Walker, & Dziuban, 2013).

OpenDSA gives us an opportunity to study the use of mobile device for learning in CS courses. We looked at logged data to get a sense of the proportion of interactions originating from mobile devices (smartphone, small size tablet, full size tablet). Since more students own tablets (handheld devices with screen size above 7"), we are interested in measuring the use of mobile device to access online tutorials. During Spring 2014, in addition to analyzing the logged data, we surveyed students in C5 to uncover the factors that encouraged or prevented the students from using their mobile device. In C5, students had to complete weekly OpenDSA assignments, and OpenDSA homework accounted for 2% of the total course grade.

Most students enrolled in courses using OpenDSA own a mobile device in addition to their Laptop/PC. We tracked the origin of user interactions between Fall 2013 and Spring 2014, and found that less than 1.5% of the traffic originated from a mobile device.

We surveyed students enrolled in C5 to gather their experiences and opinion about mobile devices and OpenDSA. We got a total of 51 responses. They revealed that:

- 68% of respondents own a laptop computer with no touch screen,
- 49% own a Touch screen laptop,
- 72% own a smartphone,
- 17% own a small tablet (approx. 7"),
- 15% own a Full-size tablet (approx. 10").

The mobile devices ownership distribution in C5 is consistent with the general US population statistics,[1] and can be generalized at least to the other VT courses that we studied.

From the log data, we found that out of 176 students enrolled in C5, only 5 accessed OpenDSA from a mobile device at least one time. The ratio of mobile users was equally low in other VT courses using OpenDSA. We did not expect smartphone owners to access the materials via their devices, because several factors make it difficult to effectively use them to study (screen size, battery life, and usability concerns). But even small- and full-sized tablet owners rarely tried to access the tutorials from their devices. This situation seemed to contradict previous study showing students positive attitude towards using full size tablets to study (Rossing, Miller, Cecil, & Stamper, 2012).

We tried to determine if the low use of mobile devices was due to some serious usability bug within OpenDSA. When asked to ranked course resources by positive learning impact, students ranked OpenDSA as the second most important (after programming labs), thus proving that the students did not have a negative opinion of the OpenDSA system. The low number of mobile users seems to indicate that the reason not to use mobile devices was made independently of OpenDSA, since the vast majority (97%) of students never accessed OpenDSA from a mobile device.

We then sought to see if the nature of the course made it harder for students to use mobile devices. We asked students if they ever used their mobile devices to study in other courses. We got 15 positive answers out of 51 responses. But there were only 2 positive answers when asked if they used mobile devices to access OpenDSA content. Further investigation will be necessary to understand the difference the difference. The most common reason mentioned by students to justify their preference for non-mobile devices was the "convenience". Most students said that it is more convenient to use a laptop or a desktop computer when working on OpenDSA assignments. They mentioned the bigger screen and the full keyboard of laptops/desktops as key elements, confirming the findings in Rossing et al., 2012. As we mentioned earlier, many

[1] http://www.pewinternet.org/factsheets/mobile-technology-fact-sheet/ (accessed 09/13/2014).

Please cite this article in press as: Fouh, E., et al. Exploring students learning behavior with an interactive eTextbook in Computer Science courses. *Computers in Human Behavior* (2014), http://dx.doi.org/10.1016/j.chb.2014.09.061

OpenDSA exercises are proficiency exercises, suitable for touch interaction, and do not require typing text. Even the KA exercises do not require the user to type in significant text.

We saw earlier that most students are "procrastinators", they wait until late to do their homework. Several studies have shown that procrastinators experienced greater stress especially toward the end of the assignment period when they are rushing to complete the assignment on time (Tice & Baumeister, 1997). For procrastinator it makes no sense to include a mobile device into the study workflow. They will go for the device that will give them more capabilities (bigger screen, possibility to open multiple tabs, full keyboard, etc.) to complete the homework in a small amount of time. Under those circumstances, mobile devices cannot compete with laptops/desktops.

6. Conclusion

We analyzed the behavior of CS students when using the OpenDSA eTextbook system. We found that a majority of students skip directly to the exercises without reading the text, which indicates that the existing text is not engaging enough or is too overwhelming for students. It is also possible that since most students waited until the due date to do the homework, reading the text was thought to "slow them down" so that they might not be able to submit the homework on time. However, perhaps seemingly in contradiction to this finding, some students voluntarily completed additional exercises as a study aid when preparing for the exams.

When students were not given credit to complete slideshows, they completed fewer than the students who were given credit. However, all of the students used slideshows to some extent, and overall use was fairly high. Skipping to the end of slideshows occurs more often when completing slideshows is part of the scoring; however, the behavior was still observed even when students did not receive credit.

The low use of mobile device might be explained by a combination of structural and behavioral factors. Structural factors include device characteristics and some usability issues, especially for small screen size devices. There was a lot of procrastination, thus leading to limited amount of time to complete homework. It is possible that this has an influence on the device selected.

Acknowledgement

This work is supported by the United States National Science Foundation, under Grants DUE-1139861 and IIS-1258471.

References

Dahlstrom, E., Walker, J. D., & Dziuban (2013). ECAR study of undergraduate students and information technology, 2013: EDUCAUSE Center for Analysis and Research.

Fouh, E., Karavirta, V., Breakiron, D. A., Hamouda, S., Hall, S., Naps, T. L., et al. (2014). Design and architecture of an interactive eTextbook – The OpenDSA system. *Science of Computer Programming, 88*, 18.

Karavirta, V., & Shaffer, C. A. (2013). JSAV: The JavaScript algorithm visualization library. In *Paper presented at the conference on innovation and technology in computer science education (ITCSE)*, Canterbury, UK.

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

Roschelle, J. (2003). Keynote paper: Unlocking the learning value of wireless mobile devices. *Journal of Computer Assisted Learning, 19*(3), 260–272. http://dx.doi.org/10.1046/j.0266-4909.2003.00028.x.

Rossing, J. P., Miller, W. M., Cecil, A. K., & Stamper, S. E. (2012). iLearning: the future of higher education? Student perceptions on learning with mobile tablets. *Journal of the Scholarship of Teaching and Learning, 12*(2), 1–26.

Shaffer, C. A., Karavirta, V., & Naps, T. L. (2011). OpenDSA: beginning a community Hypertextbook project. In *Paper presented at the Koli Calling international conference on computing education research*, "Koli National Park, Finland.

Shaffer, C. A., Naps, T. L., & Fouh, E. (2011). Truly interactive textbooks for computer science education. In *Paper presented at the sixth program visualization workshop*, Darmstadt, Germany.

Tice, D. M., & Baumeister, R. F. (1997). Longitudinal study of procrastination, performance, stress, and health: The costs and benefits of dawdling. *Psychological Science, 8*(6), 454–458. http://dx.doi.org/10.2307/40063233.

Wu, W.-H., Jim Wu, Y.-C., Chen, C.-Y., Kao, H.-Y., Lin, C.-H., & Huang, S.-H. (2012). Review of trends from mobile learning studies: A meta-analysis. *Computers & Education, 59*(2), 817–827. http://dx.doi.org/10.1016/j.compedu.2012.03.016.