Supporting Active Learning Using an Interactive Teaching Tool in a Data Structures and Algorithms Course

Tommy Färnqvist*, Fredrik Heintz*, Patrick Lambrix*, Linda Mannila† and Chunyan Wang*
*Department of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden
Email: {firstname.lastname}@liu.se
†Department of Information Technologies
Åbo Akademi, Joukahaisenkatu 3-5 A, FIN-20520 Turku, Finland
Email: linda.mannila@abo.fi

(Extended Abstract)

***Abstract*—Traditionally, theoretical foundations in data structures and algorithms (DSA) courses have been covered through lectures followed by tutorials, where students practise their understanding on pen-and-paper tasks. In this paper, we present findings from a pilot study on using the interactive e-book OpenDSA as the main material in a DSA course. The goal was to redesign an already existing course by building on active learning and continuous examination through the use of OpenDSA. In addition to presenting the study setting, we describe findings from four data sources: final exam, OpenDSA log data, pre- and post course questionnaires as well as an observation study. The results indicate that students performed better on the exam than during previous years. Students preferred OpenDSA over traditional textbooks and worked actively with the material, although a large proportion of them put off the work until the due date approaches.**

***Index Terms*—Computer science education, computer aided instruction**

I. Introduction

TRADITIONALLY, covering the theoretical foundations in a data structures and algorithms (DSA) course uses lectures followed by tutorials, where students practise their understanding on pen-and-paper tasks. In these courses, however, many concepts benefit from computer based visualisation. Since the 1960s a multitude of tools and activities using visualisation have been developed for DSA courses, while didactic research investigating how to use such tools has become more common. Until recently, though, there was no all-in-one way to integrate these tools into a traditional DSA course. Now, the open project OpenDSA [1], [2], aims at providing such an integrated tool. Some of its main features are the following. OpenDSA is an e-learning tool that provides textbook quality text with links to a dictionary of important terms. It provides many code examples and step-by-step algorithm visualizations. Understanding of algorithms is supported by simulation exercises for the algorithms. After each section and each chapter, there are quizzes with multiple choice questions related to the main concepts of the section or chapter. Common to all examples, exercises and quizzes is that they are randomly generated instances of examples and exercises, thereby providing for a multitude of practise possibilities. As a help for doing the exercises and quizzes, students can obtain hints, and automated and immediate feedback is given. Further, OpenDSA automatically stores extensive log data about the interaction with the system. For more details on how OpenDSA works and its features we refer to Fouh et al. [1].

In this paper, we present findings from a pilot study on using OpenDSA as the main material in a course on data structures and algorithms. We wanted to design a setting that builds on two main pedagogical principles: (1) *active learning* [3], where we, by employing a blended learning environment, could give students more responsibility and control over their own learning; and (2) *continuous examination* with individual and immediate feedback. We begin by briefly describing the rationale and pedagogical underpinnings for using OpenDSA in the course. Next, we present our study and the methods and data collection instruments used, after which we discuss the results. The paper is concluded with some ideas for future work. Due to the limited space available, a lot of details concerning how the study was conducted and detailed results have been left out in favour of the concluding discussion. We refer the interested reader to the upcoming SIGCSE paper [4].

II. Pedagogical Rationale

Active learning [3] builds on constructivist principles, according to which students become active participants in their own learning process [5]. Instead of viewing learning as passive transmission of information from the teacher to the students, learning is considered an active process, in which students themselves construct the knowledge by building further upon their prior knowledge. Moreover, both constructivism and active learning are related to the cone of experience developed by Dale in the 1940s. This model suggests that students retain more information by what they do (90%) compared to what they read (10%), hear (20%) or observe (30%). Consistent with this model, Ramsden [6] suggests that most students cannot learn unless they are actively involved.

In addition to being actively involved in their studies, students also need feedback on their work – learning requires practice with frequent feedback [7]. To stop students from

developing misconceptions that hinder learning it is important to discover and address them quickly [7]. Visualizations and interactive exercises have potential to increase student engagement [8], and material that is perceived as engaging also drives students to invest more effort in studying it [9].

Lately, interactive and electronic study material has become increasingly popular, partly through the introduction of massive open online courses (MOOCs). Students are provided continuous access to different types of material, examples and exercises. From a teacher and research perspective, this kind of computer based, interactive material makes it possible to gather large amounts of data, ranging from time stamped interactions (e.g., page loads and button clicks) to performance on different types of tasks [10]. These data can be used to gain insight into student activity and learning.

The interactive exercises in OpenDSA engage students and give them increased opportunities for practicing their knowledge through practical hands-on work (analyzing problems, working with varying types of data structures, as well as applying algorithms to different structures). Furthermore, the summary questions at the end of each chapter give students a way of forming their own opinion on how well they succeeded in grasping the current concept. This is needed since learners tend to overestimate how much they have learned [11].

III. Our Study

A. The Study Setting

At the start of their second year, the CS majors at Linköping University used to take a DSA course (TDDC70) comprising 6 ECTS credits. TDDC70 was a traditional monolithic course, with lectures, pen-and-paper based tutorials, and programming assignments. The course was assessed via a written exam at the end of the course for 4 credits, graded (fail, 3, 4, 5), and programming assignments for 2 credits, graded pass/fail. Since 2014 they instead take an 11 credit course (TDDD86) where the DSA contents is the same as in TDDC70 (with a few smaller additions), with the rest of the credits spent on teaching C++ and a small introduction to programming paradigms. It is in the TDDD86 course we have deployed OpenDSA.

Within the TDDD86 course we tried to discern how to best use the system and changes to the organisation of the course to create a blended learning environment supporting an active learning style among the students by giving the students more responsibility for and control over their own learning as well as employing continuous examination with individualised feedback.

One of the important criteria when designing the new version of the DSA course was to give students the opportunity to work at their own, albeit steady, pace, through continuous individual examination with immediate feedback. The mechanisms for this are already in place within OpenDSA, but to give the desired change more support we also changed the examination of the course. Originally, the theoretical content of TDDD86 was supposed to be assessed by a written exam, comprising 5 ECTS credits, at the end of the course. We instead opted to split the exam credits in two parts: a two credit part for completing all assigned modules and exercises in OpenDSA, and a three credit part for an exam given at the end of the course. The OpenDSA exercises were to be completed by the students during non-scheduled time (compared to TDDC70, where tutorials were scheduled for doing paper-and-pen exercises.) The new exam consisted of one computer based part (using a small sample of the OpenDSA exercises used during the course) for which completion gave a passing grade (3) on the exam. For a higher grade (4, 5) the students needed to solve written problems testing deeper understanding of and connections between different DSA concepts.

B. Data Collection and Methodology

Our study involves data from four different sources; exam results, log data captured by the OpenDSA system, questionnaires, and an observation study. These are briefly described in the following:

1) Final Exam: We chose to include final exam results for TDDC70 from 2010–2013, since these had the same examiner as in TDDD86. Also, the written problems for a higher grade on the TDDD86 final exam were of the same types and estimated difficulty as the hardest types of problems used in the TDDC70 final exams.

2) Log Data: OpenDSA automatically stores extensive log data covering all interactions with the system. Hence, the data makes it possible to study aspects related to student behavior (how and when they study), student learning (performance on different types of exercises) and technical aspects (platforms and browsers used). A thorough overview of all data stored can be found on the OpenDSA web site (http://algoviz.org/OpenDSA/Doc/manual/DatabaseSchemas.html#exercise). The data can be used to investigate very specific issues related to these three areas. In this study, we have focused on log data related to student activity – how often they have used the material and at what times.

3) Questionnaires: To study the experience and attitude among the students toward interactive textbooks in general and our material specifically we conducted two online surveys. The first survey was distributed electronically to all the students at the start of the course and second survey after the end of the course. The first survey was answered by 54 of 130 students (42%), while 35 students (27%) the second survey. Our experience from other studies is that it is harder to get students to answer a questionnaire after the end of the course, there are of course also fewer active students as some have dropped out.

4) Observation study: We decided to use an observation study as it allows us to gather data in the specific setting to be studied [12], which in our case is that the students use the interactive book for studying. We used two observers which were nonparticipant observers. One of the observers has domain knowledge (former teacher in the area) while the other observer has no domain knowledge (language teacher). We expected the observers to focus on different things. The observers took descriptive as well as reflective notes.

Six students, five males and one female, participated in the observations. They were paid volunteers. For privacy reasons,

in the remainder of this paper we refer to each student participating in the observation study as 'he'.

The students were each observed individually three times. In each session, the students worked on the chapters selected by the examiner and the observers together. The first observation was at the beginning of the term. The aim was to let the students become familiar with the observation. The chapter for the first session contained a large amount of text. The second session was arranged mid-term, while the third and final session was at the end of the term. The chapters for the second and third session required the students to be active and interact with the material. For the second and third session the students were asked not to work on the respective chapters before the sessions.

IV. Findings and Discussion

In Table I we can see the results on the final exam for different years in TDDC70 and TDDD86, respectively. Statistical calulations reveal that our experiment did not influence the average (passing) grade that much, but if we look at the proportion of students getting a higher than passing grade out of all students taking the exam, we have a significant increase when compared to the exams from 2010, 2012, and 2013 ($p < 0.02$, when looking at results from 2-sample χ^2-tests for equality of proportions). We also note that the 2011 TDDC70 exam has similar properties when compared to the other TDDC70 exams, but with a lower significance ($p < 0.05$).

	fail	3	4	5	avg	prop
TDDC70 2010	39	29	10	6	3.48	19 %
TDDC70 2011	19	42	21	3	3.41	28%
TDDC70 2012	59	25	8	2	3.34	10 %
TDDC70 2013	64	27	11	1	3.33	12 %
TDDD86 2014	0	71	30	15	3.51	39 %

TABLE I
NUMBER OF STUDENTS RECEIVING DIFFERENT GRADES, AVERAGE PASSING GRADE, AND PROPORTION OF STUDENTS RECEIVING A HIGHER THAN PASSING GRADE OUT OF ALL STUDENTS TAKING THE EXAM.

When looking at final exam results, one thing that stands out is that no students failed the exam in TDDD86, while quite a number of students failed the TDDC70 exams. Is this because the TDDD86 exam was easier? In a certain sense this is probably true, but at the same time it is a fact that 117 students finished all assigned OpenDSA exercises, and out of these 116 took the exam — meaning they were very well prepared for the OpenDSA part of the final exam. As we have no way of measuring how much drilling the TDDC70 students had done on the "easier" type of questions testing definitions and understanding of basic properties it is hard to make a fair comparison, but since the OpenDSA exercises were mandatory in TDDD86 and the students could work in their own pace continously throughout the course, there is probably a big change in how much time the students have spent learning the basic concepts.

We have also seen, that there seems to be a change in how many students aspire for and make a higher grade on the final exam. Is this due to the fact that the TDDD86 students might have done more drilling than their TDDC70 counterparts? We certainly hope so — being more familiar with the properties of various data structures and their algorithms should make you better prepared to answer the harder questions requiring insight into how to combine different structures to solve algorithmic problems. However, we cannot rule out the possibility that the change in exam results is just due to the student group. To control for this we intend to make an analysis factoring in performance also on previous courses. A further confounding factor stems from other course acitivites; in TDDC70 the students did four mandatory programming assignments, while TDDD86 has eight mandatory programming assignments out of which six has clear algorithmic content. It could be the case that the higher exposure to solving more complex problems made the TDDD86 students significantly better prepared for those types of questions than the TDDC70 students. On the other hand, using OpenDSA enabled the examiner of the TDDD86 course to skip some lecture time used in TDDC70 to provide detailed demonstrations of data structure behaviour, leaving the TDDD86 students to do that kind of course work on their own, meaning that the TDDD86 students actually had less DSA lectures scheduled than the TDDC70 students.

The log data suggest that students actively used the material throughout the course. Having all students submitting on average solutions to nearly 90 exercises is a big improvement to when using traditional material with a limited number of exercises to practice on. Students have also used OpenDSA continuously throughout the day and work week, suggesting that students have used it also outside of the university (See Figure 1).

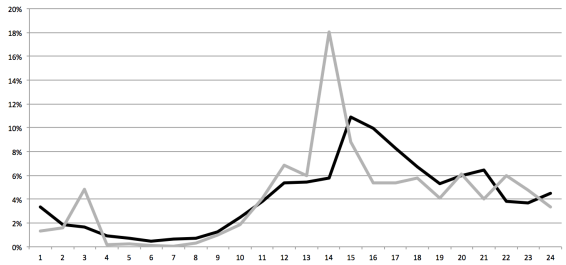

Fig. 1. Hourly distribution of student activity with OpenDSA throughout the course. All exercise submissions (black) and overall interactions (grey) throughout the 24 hours in a day.

Nevertheless, the monthly and daily distribution of submissions (Figure 2 and 3) show that most activity took place around the final exam and the due date for submitting compulsory exercises for credit. This is in line with the findings of Fouh [13], who found that students using OpenDSA tend to wait until the deadline before dealing with the tasks assigned. One thing to consider is hence, whether a more continuous formative assessment scheme, using, for instance, both OpenDSA exercises and a learning journal, could aid in helping students distribute their work more evenly.

The findings from the questionnaires show that the overall student experience with OpenDSA is very positive, 89% satisfied and 91% prefer OpenDSA over printed textbook, and the only issue is bugs. One student writes in a free text answer: 'When you get rid of the bugs this will be one of the best text

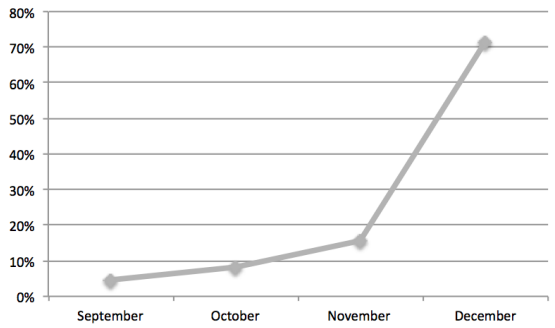

Fig. 2. Monthly distribution of exercise submissions (not including exercises on the exam)

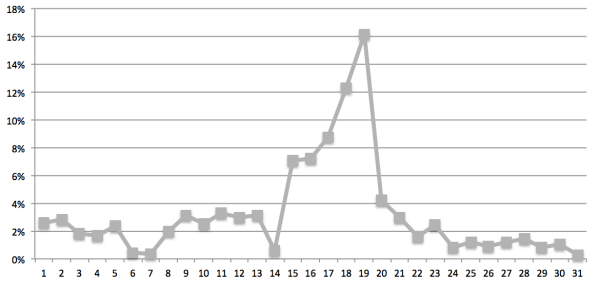

Fig. 3. Daily distribution of exercise submissions in December (not including exercises on the exam)

books you can get. Animations in combination with text is very close to what you get from a good lecture." The students also tend to spend more time with the material, can work in their own pace and keep a steady tempo through the course. Some comments from the students regarding this: "I have learned more in less time, [...] [I] can go through the material in the pace that is needed.", "The way the exercises were presented in OpenDSA made me spend more time with them after reading the chapter compared to a printed textbook, which is positive!" and "Since there were mandatory exercises, and quite many, I read and did the exercises in the same pace as the lectures, which made it easier to study for the exam at the end and I had more knowledge and a better understanding."

Some main positive findings from the observation study are that (1) students prefer the interactive book over a traditional book, (2) the dictionary, visualizations and exercises (the interactive parts of the book) are appreciated and help students understand, and (3) the interactive book encourages students to work during the course.

The observations also revealed some issues that require special attention when using an interactive book: there are students that (1) skip text, (2) guess answers to exercises and (3) do the interactions in the visualizations without learning.

V. Future work

The findings presented in this paper indicate that the initial experience from the redesigned version of the DSA course, incorporating OpenDSA as the main material, has been positive. As stated above, OpenDSA collects a large amount of interesting data making it possible to study a multitude of interesting questions. Future work involves investigating, for instance, potential relationships between student activity in OpenDSA and exam performance. Also, the TDDD86 course is now (fall 2015) offered again using the same format, hence adding to the data available for analysis.

We are also interested in investigating how the log data can be used to guide the learning and the teaching in a dynamic way during the course, so called Just-in-Time Teaching [14]. For instance, for topics that from the log data are identified to be difficult, extra sessions could be scheduled.

Acknowledgments

The authors want to acknowledge the timely and competent support provided by Cliff Shaffer and the OpenDSA team making this study possible. The study was partially financed by funding for pedagogical projects at Linköping University, and by the pedagogical development group (PUG) at the Institute of Technology at Linköping University.

References

[1] E. Fouh, V. Karavirta, D. A. Breakiron, S. Hamouda, S. Hall, T. L. Naps, and C. A. Shaffer, "Design and architecture of an interactive etextbook - The OpenDSA system," *Science of Computer Programming*, vol. 88, pp. 22–40, 2014, software Development Concerns in the e-Learning Domain.

[2] C. A. Shaffer, V. Karavirta, A. Korhonen, and T. L. Naps, "OpenDSA: Beginning a community active-eBook project," in *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '11, 2011, pp. 112–117.

[3] C. C. Bonwell and J. A. Eison, *Active learning: Creating excitement in the classroom*, ser. ASHE-ERIC Higher Education Report No. 1. George Washington University, 1991.

[4] T. Färnqvist, F. Heintz, P. Lambrix, L. Mannila, and C. Wang, "Supporting active learning by introducing an interactive teaching tool in a data structures and algorithms course," in *Proceedings of the 47th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '16, to appear.

[5] M. Ben-Ari, "Constructivism in computer science education," *SIGCSE Bull.*, vol. 30, no. 1, pp. 257–261, Mar. 1998. [Online]. Available: http://doi.acm.org/10.1145/274790.274308

[6] P. Ramsden, *Learning to Teach in Higher Education*. Routledge, 1992.

[7] J. D. Bransford, A. L. Brown, and R. R. Cocking, Eds., *How People learn: Brain, mind, experience, and school*. National Academy Press, 1999.

[8] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of algorithm visualization effectiveness," *Journal of Visul Languages & Computing*, vol. 13, no. 3, pp. 259–290, 2002.

[9] J. Sorva, V. Karavirta, and L. Malmi, "A review of generic program visualization systems for introductory programming education," *ACM Transactions of Computing Education*, vol. 13, no. 4, pp. 15:1–15:64, 2013.

[10] A. Korhonen, T. Naps, C. Boisvert, P. Crescenzi, V. Karavirta, L. Mannila, B. Miller, B. Morrison, S. H. Rodger, R. Ross, and C. A. Shaffer, "Requirements and design strategies for open source interactive computer science ebooks," in *Proceedings of the ITiCSE Working Group Reports Conference on Innovation and Technology in Computer Science Education-working Group Reports*, ser. ITiCSE -WGR '13. New York, NY, USA: ACM, 2013, pp. 53–72. [Online]. Available: http://doi.acm.org/10.1145/2543882.2543886

[11] R. A. Bjork, J. Dunlosky, and N. Kornell, "Self-regulated learning: Beliefs, techniques, and illusions," *Annual Review of Psychology*, vol. 64, pp. 417–444, 2013.

[12] J. W. Creswell, *Educational Research: planning, conducting, and evaluating quantitative and qualitative research*, 4th ed. Pearson, 2012.

[13] E. Fouh, D. A. Breakiron, S. Hamouda, M. F. Farghally, and C. A. Shaffer, "Exploring students learning behavior with an interactive etextbook in computer science courses," *Computers in Human Behavior*, vol. 41, pp. 478–485, 2014.

[14] T. Bailey and J. Forbes, "Just-in-time teaching for CS0," *SIGCSE Bulletin*, vol. 37, no. 1, pp. 366–370, 2005.