Effectiveness of Algorithm Visualisation in Studying Complex Algorithms: A Case Study using TRAKLA2

Ravie Chandren Muniyandi[1*], Ali Maroosi[2]

[1, 2] Research Centre for Software Technology and Management,
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia,
43600 Bangi, Selangor, Malaysia
ravie@ukm.edu.my, ali.maroosi@gmail.com
[*]Corresponding Author

ABSTRACT

Algorithm visualisation (AV) can be utilised to improve students' programming and programme comprehension skills. Visual tools motivate students and teachers to more effectively study or teach complex algorithms. This study demonstrates that AV tools significantly improve student skills and scores and enhance understanding of complex algorithms to a degree greater than those of less complexity. TRAKLA2 is a visualisation tool used to enhance the process of learning algorithm construction and optimization. To assess the effectiveness of TRAKLA2, students were given an algorithm test prior to being introduced to the software. Students then used TRAKLA2 to learn five different types of algorithms. Analysis of students' feedback from questionnaires and tests showed greater levels of improvement in understanding complex problems as compared to those of less complexity. Our results show that TRAKLA2 enhances the understanding of complex algorithms and is an effective tool for algorithm teaching and learning.

Keywords: algorithm visualisation, algorithm learning, complex algorithms, TRAKLA2, MatrixPro

INTRODUCTION

Students often have significant problems learning basic programming concepts. Many educators agree that teaching programming to novice is a complex and difficult process (Robling 2010; Lahtinen, 2006). The problems arise from both lack of teaching resources and large class sizes that often prevent adequate personal instruction (Tuparov et al., 2012). Subsequently, there is a need for instructional tools that support independent student learning (Rajala et al., 2009). One method tested as a means to help students learn basic programming concepts is software visualisation (Nikander et al., 2010; Schoeman et al., 2013). This branch of software engineering uses graphics and animation to illustrate different aspects of computing (Nikander et al., 2010; Stasko et al., 1998; Price et al., 1993) and is divided into two sub categories: programme visualisation and algorithm visualisation (AV). Programme visualisation uses various visual techniques to enhance the students' understanding of computer programmes and is typically used to illustrate actual, implemented programmes. AV illustrates algorithmic concepts as abstractions and is independent of any actual algorithm implementation.

There are very few studies on the effectiveness of AV tools. Thus, it is important to examine how visualisations can be utilised to help teachers plan which tools to use, when to use them, and for what topics (Rajala et al., 2011). Algorithms play a central role in many areas of computing, requiring students to become familiar with a wide range of examples (Robling et al., 2011).

There are many software systems used to visualise algorithms (Nikander et al., 2011). GASP (Shneerson et al., 2000; Tal et al., 1994), GAWAIN (Hausner et al., 1998), and Vega (Hipke et al., 1998) illustrate geometric algorithms, while Hull2VD (Fisher, 2004) and VoroGlide (Icking, n.d.) provide visualisations for Voronoi diagrams (Aurenhammer, 1991) and GeoWin (Basken, 2002) is the visualisation tool specifically for the LEDA algorithm. TRAKLA2 is an AV tool capable of automatically assessing exercises (Nikander et al., 2010). Studies show that it enhances student motivation to learn and understand algorithms when properly engaged in a learning task (Rajala, 2010). TRAKLA2 is a learning environment that utilises visual algorithm simulation to deliver tracing exercises to students

(Malmi et al., 2004). The system automatically assesses students' solutions and provides feedback on the correctness of the simulation. There are two major differences between the visualisations included in TRAKLA2 and previous systems (Nikander et al., 2009). First, TRAKLA2 exercises are designed to be used in teaching geoinformatics, while other systems are designed primarily for teaching general computer science. Second, the tracing exercises offer a different type of interaction to other systems (Nikander et al., 2009). The students solve problems by constructing their own algorithm animation sequences using visual interaction. This occurs at the level of constructing the engagement taxonomy (Naps et al., 2003). Therefore, unlike other systems, algorithm creation is not required prior to constructing a new animation.

In this study, we applied the learning environments associated with TRAKLA2 and MatrixPro (Karavirta et al., 2004) (which are based on the same core as TRAKLA2) to courses teaching data structures and algorithms. Our results indicate an improvement in student comprehension of complex algorithms.

PREVIOUS WORKS

TRAKLA2 is among the most widely used AV tools. The program allows the learner to control the visual representation of the data structures manipulated by the algorithm and to build data structures by dragging and dropping their elements through the use of a graphical user interface. TRAKLA2 exercises require learners to choose a series of operations that will alter the state of the data structure to achieve a particular outcome. For example, students might build a tree-type data structure by repeatedly dragging new values to the correct locations in the tree. Alternatively, the learner can practise and gain understanding of an algorithm by examining its step-by-step execution to produce a model solution (Fouh et al., 2012).

In 2002, Korhonen et al., carried out the first intervention study with an earlier version of TRAKLA2 using three randomised student groups to compare final examination results between students completing instructed simulation exercises in a classroom session and those is using a web-based learning environment. The results showed that if the exercises were the same, there was no significant difference in examination results between students who had undertaken exercises on the web or in the classroom. However, in cases where the exercises were more challenging, there was a significant difference in the results.

TRAKLA2 was introduced at the University of Turku by Laakso et al., 2005. The learning capacity of students using TRAKLA2 during classes teaching data structures and algorithms was compared to those not using the software. Additionally, survey data were collected tracking changes in students' attitudes towards web-based learning environments. The results showed that TRAKLA2 increased positive attitudes toward web-based learning. According to student self-evaluations, the best learning outcomes were achieved by combining traditional and web-based exercises. Furthermore, overall student performance improved compared to results from 2003, when instruction was undertaken using only traditional methods.

In 2007, Myller et al. conducted an experimental study focusing on engagement taxonomy. The learning outcomes were compared with students taught using visualisation on different engagement levels. The results indicated that higher levels of engagement had a positive effect on students' results in favour of the intervention group, especially where students without previous knowledge of data structures or algorithms were concerned.

In 2009, Laakso et al. studied the effects of AV on collaborative teaching and learning. The use of visualisations for collaborative learning introduced new challenges for visualisation tools. Pre- and post-tests were used as instruments in the experiment. No statistically-significant differences were found in post-testing between randomised groups and not all of the students that were assigned participated at the engagement level. Students also did not solve TRAKLA2 exercises, but instead only watched the model solution. By regrouping the students based on the monitored behaviour, differences were discovered in the total and pair average based on the post-test scores.

TRAKLA2 has also been used for teaching spatial data algorithms (Nikander et al., 2009). The study analysed students' progress using qualitative methods to discover how the new system altered learning outcomes. TRAKLA2 use increased examination scores associated with learning spatial data algorithms.

PROBLEM STATEMENT

There is growing interest in evaluating the educational impact of AV tools, with many results indicating their positive effects on the learning process. The effectiveness of AV methods depends on the appropriate use of AV tools. Studying AV effectiveness based on problem complexity provides valuable information to lecturers and students regarding the necessary time constraints associated with using AV tools. The impact of problem complexity on the effectiveness of AV tools has not been discussed in previous research. This study investigates the relationship between problem complexity and the effects of using AV tools to enhance the ability of students to solve complex problems. The aims of this study can be summarized as follows:

1. Confirm that AV-based tools enhance the learning process associated with data structures and algorithms.

2. Show that AV-based tools enhance students' ability to solve complex algorithmic problems as compared to problems of lesser complexity.

Research Design

An experiment was conducted to evaluate the ability of TRAKLA2 to enhance the ability of students to learn problems of varying complexity. The research aims to answer two questions: 1) Does TRAKLA2 help students to learn data structures and algorithms? 2) Is there a difference in how TRAKLA2 helps students learn complex problems compared to problems of lesser complexity?

The experiment was conducted in the data structures and algorithms course at the Universiti Kebangsaan Malaysia (UKM) during the first semester of the 2013-2014 academic year. Students were taught algorithms conventionally without using any visual software. Three weeks prior to testing, students were instructed to prepare for a test on binary search (basic algorithms), insertion sort and merge sort (sorting algorithms), and depth-first search (DFS) and breadth-first search (BFS) (search trees). Following this test, students were introduced to TRAKLA2 and given 2 weeks to prepare for a second test on the same topics.

The attitudes of the UKM students were evaluated through a questionnaire administered following the second test. A second questionnaire was administered to evaluate whether TRAKLA2 enhanced learning problems of greater or lesser complexity, the extent to which TRAKLA2 helped students learn the algorithms being tested, and the amount of time students spent learning the topics. Students ranked the topics according to how helpful TRAKLA2 had been, as well as how the topics were designed, developed, and explained.

Methods

The experiment included two tests containing questions of varying complexity and a questionnaire to evaluate student attitudes toward the usefulness of TRAKLA2. Students were examined alone during the experiment and while taking both tests. The style of questions for both tests was identical. The only differences in testing parameters involved the students' ability to practise problems using TRAKLA2 prior to the second test and the questionnaires administered to students after using TRAKLA2 for the specified topics.

Participants

The Faculty of Information Science and Technology at UKM has been offering courses on Data Structure, and Analysis and Design of Algorithms for undergraduate computer science students in order to teach the skills of algorithm analysis and design techniques. The courses cover the main concepts and principles of data structures and algorithm design and analysis. The experiment was undertaken with a class of 32 undergraduate students.

Materials

Both tests consisted of the same seven questions, with each question having multiple steps scored individually. In each question, the students were presented a code fragment and asked to define the output or the state of the programme. The first three questions presented pseudocode representing a binary search algorithm and asked the student to describe the steps involved in finding a number from an array of numbers using the algorithm. The fourth question presented pseudocode describing an insertion sort algorithm and asked the student to arrange an array of numbers using the algorithm and describe the contents of array variables throughout the sorting process. The sixth question presented pseudocode representing a DFS algorithm and asked the student to describe the process of exploring nodes in a graph and annotate into a table the order in which each node was visited. The seventh question asked the student to explain the process associated with a BFS algorithm using the same format as described in the sixth question.

A questionnaire was administered to students following the second test in order to evaluate their attitude regarding the effectiveness of TRAKLA2. The questions examine their levels of progress in each subject based on the complexity of the problems and the time required to obtain higher scores in the second test as compared to the first. Some examples of questions are as follows: "What was your skill level before using TRAKLA2?"; "What is your skill level after using TRAKLA2?"; "How many example problems were solved using TRAKLA2 to obtain the highest possible score?"; "How long did it take to practice the problems using TRAKLA2?"; "Rank the effectiveness of TRAKLA2 for learning five algorithms: (1) Binary Search (2) Insertion Sort (3) Merge Sort (4) DFS (5) BFS".

PROCEDURES

The study was performed in the final 3 weeks of the first semester of the 2013-2014 academic years. One week prior to the first test, students were advised to prepare for examination on five subjects taught during the semester. The time reserved for the examination was 120 minutes. Following the first test, each student independently used TRAKLA2 for 2 weeks prior to the second test. The highest possible score for both tests was 100%. The questionnaire was completed by students independently after practising for 2 weeks using TRAKLA2. Finally, comparisons of the results from each test were used to evaluate student progression, with questionnaire responses used to further validate the test results.

RESULTS AND DISCUSSIONS

The results indicate that TRAKLA2 effectively increased student comprehension of complex algorithms compared to those of lesser complexity. In the first test, the students' mean score was 40%. In the second test, after practising with TRAKLA2, the mean score increased to 75.4%. This illustrates the effectiveness of TRAKLA2 as an AV tool. Interestingly, student progress in each topic varied according to problem complexity.

In the first test, the scores for (i) binary search, (ii) insertion sort, (iii) merge sort, (iv) DFS, and (v) BFS were 50%, 35%, 80%, 20%, and 20%, respectively. Following TRAKLA2 use, the scores increased to 80%, 75%, 100%, 65%, and 65%, respectively. The increases were 30%, 40%, 20%, 45%, and 45%, respectively.

The feedback from the questionnaire (Table 1) validates the observed improvement in test scores. Students reported that their skills in all topics increased following TRAKLA2 use (Table 1 and Figure 1). The perceived skills of the students improved for binary search by 25%, insertion sort by 30%, merge sort by 5%, DFS by 35%, and BFS by 40%. According to questionnaire results, it is possible to rank TRAKLA2 effectiveness regarding the increase in student skill as follows: 1. BFS, 2. DFS, 3. Insertion Sort, 4. Binary Sort, and 5. Merge Sort. These results show that TRAKLA2 was most effective at improving student understanding of the most complex algorithms as compared to those of lesser complexity.

Table 1: Average Scores of Student Questionnaire Responses

	Binary search	Insertion sort	Merge sort	DFS	BFS
What was your skill level before using TRAKLA2?	70%	60%	90%	50%	45%
What is your skill level after using TRAKLA2?	95%	90%	95%	85%	85%
How many example problems were solved using TRAKLA2 to obtain the highest score?	7 times	10 times	5 times	10 times	10 times
How long did it take to practice the problems using TRAKLA2?	10 min	15 min	10 min	45 min	50 min

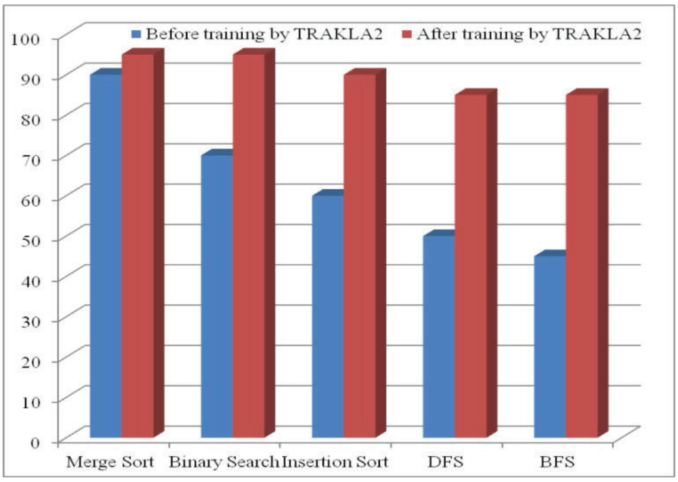

Figure 1: Comparison of Student Scores Associated with Different Algorithms Before and After Using TRAKLA2 to Prepare for Tests

CONCLUSION

In this study, two tests were administered to students on five subjects of varying complexity. The first test was administered without students having prepared using any AV tools. The second test was administered following student preparation using TRAKLA2. A questionnaire was given to students to evaluate their attitudes regarding TRAKLA2 effectiveness in helping them learn problems of varying complexity. The results indicate that TRAKLA2 is most effective at helping students to learn complex algorithms.

TRAKLA2 use increased test scores associated with questions related to less complex algorithms by 5%, as compared to a 40% increase associated with complex algorithms. Furthermore, students indicated that TRAKLA2 improved their ability to learn complex algorithms to a greater extent than those of lesser complexity, confirming the results observed from the test scores. These results illustrate the effectiveness of AV tools in teaching and learning complex algorithms.

ACKNOWLEDGEMENT

This study was supported by Universiti Kebangsaan Malaysia by an Action Research Grant (UKM-PTS-2012-133).

REFERENCES

Aurenhammer, F. (1991). Voronoi diagrams: a survey of a fundamental geometric data structure, ACM Computer Survey, 23, 345-405.

Basken, M. and Naher, S. (2002). Geowin a generic tool for interactive visualisation of geometric algorithms, Lecture Notes in Computer Science (Software Visualisation), 2269: 637-639.

Fisher, J. (2004). Visualizing the connection among convex hull, voronoi diagram and delaunay triangulation,
37th Midwest Instruction and Computing Symposium.

Fouh, E., Akbar, M. & Shaffer, C. A. (2012). The role of visualisation in computer science education, Computers in the Schools, 29(1-2), 95-117.

Hausner, A. & Dobkin, D. P. (1998). Gawain: visualizing geometric algorithms with web-based animation, SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry, 411-412.

Hipke, C. A. & Schuierer, S. (1998). Vega—a user-centered approach to the distributed visualisation of geometric algorithms, Technical report, University of Freiburg.

Icking, C., Klein, R., Kollner, P. & Ma, L. (2003) Java applets for the dynamic visualisation of voronoi diagrams, Lecture Notes in Computer Science, 2598: 191-205.

Karavirta, V., Korhonen, A., Malmi, L. & Staylnacke, K. (2004). MatrixPro: A tool for on-the-fly demonstration of data structures and algorithms, Proceedings of the Third Program Visualisation Workshop, The University of Warwick, 26-33.

Korhonen, A., Malmi, L. Myllyselkä, P. & Scheinin, P. (2002). Does it make a difference if students exercise on the web or in the classroom? Proceedings of the 7th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education. New York: ACM Press, 121-124.

Laakso, M.-J., Myller, N. & Korhonen, A. (2009). Comparing Learning Performance of Students Using Algorithm Visualisations Collaboratively on Different Engagement Levels, Educational Technology & Society, 12(2): 267-282.

Laakso, M.-J., Salakoski, T., Grandell, L., Qiu, X., Korhonen, A. & Malmi, L. (2005). Multi-perspective study of novice learners adopting the visual algorithm simulation exercise system TRAKLA2, Informatics in Education, 4(1), 49-68.

Lahtinen, E. (2006). Integrating the use of visualisations to teaching programming. Proceedings of the conference Methods: Materials and Tools for Programming Education, 7-13.

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

Myller, N., Laakso, M. & Korhonen, A. (2007). Analyzing engagement taxonomy in collaborative algorithm visualisation, Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education-ACM Press, 251-255.

Naps, T. L., Roling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodgers, S. & Velazquez-Iturbide, J. A. (2003). Exploring the role of visualisation and engagement in computer science education, SIGCSE Bulletin, 35, 131-152.

Nikander, J., Helminen, J. & Korhonen, A. (2009). Experiences on using TRAKLA2 to teach spatial data algorithms, Electronic Notes in Theoretical Computer Science, 224, 77-88.

Nikander, J., Helminen, J. & Korhonen, A. (2010). Algorithm Visualisation System for Teaching Spatial Data Algorithms, Journal of Information Technology Education, 9.

Price, B. A., Baecker, R. M. & Small, I. S. (1993). A principled taxonomy of software visualisation, Journal of Visual Languages and Computing, 4(3): 211-266.

Rajala, T., Kaila, E., Holvitie, J., Haavisto, R., Laakso, M. & Salakoski, T. (2011). Comparing the collaborative and independent viewing of program visualisations, Frontiers in Education Conference (FIE), F3G-1-F3G-7.

Rajala, T., Salakoski, T., Kaila, E. & Laakso, M. (2010). How does collaboration affect algorithm learning?: A case study using TRAKLA2 algorithm visualisation tool, 2nd International Conference on Education Technology and Computer (ICETC), 3, 504.

Rajala, T., Salakoski, T., Laakso, M. J. & Kaila, E. (2009). Effects, experiences and feedback from studies of a program visualisation tool, Informatics in Education-An International Journal, 8: 17-34.

Robling, G. (2010). A family of tools for supporting the learning of programming, Algorithms, 3(2): 168-182.

Robling, G., Mihaylov, M. & Saltmarsh, J. (2011). AnimalSense: combining automated exercise evaluations with algorithm animations, Proceedings of the 16th annual joint conference on Innovation and technology in computer science education, 298-302.

Schoeman, M., Gelderblom, H. & Muller, H. (2013). Investigating the effect of program visualisation on introductory programming in a distance learning environment, African Journal of Research in Mathematics, Science and Technology Education, 17(1-2): 139-151.

Shneerson, M. & Tal, A. (2000). Interactive collaborative visualisation environment for geometric computing, Journal of Visual Languages & Computing, 11: 615-637.

Stasko, J. T., Domingue, J. B., Brown, M. H. & Price, B. A. (1998). Software visualisation: Programming as a multimedia experience, MIT Press.

Tal, A. & Dobkin, D. (1994). Gasp-a system for visualizing geometric algorithms, Proceedings of IEEE Conference on Visualisation, 149-155.

Tuparov, G., Tuparova, D. & Tsarnakova, A. (2012). Using interactive simulation-based learning objects in introductory course of programming, Procedia-Social and Behavioral Sciences, 46: 2276-2280.