Korhonen, Ari; Helminen, Juha; Karavirta, Ville; Seppälä, Otto

TRAKLA2

Published in:
9th Koli Calling International Conference on Computing Education Research, Koli, Finland, October 29 November 1, 2009

Published: 01/01/2010

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Korhonen, A., Helminen, J., Karavirta, V., & Seppälä, O. (2010). TRAKLA2. In A. Pears, & C. Schulte (Eds.), *9th Koli Calling International Conference on Computing Education Research, Koli, Finland, October 29 November 1, 2009* (pp. 43-46). University of Joensuu. http://www.it.uu.se/research/publications/reports/2010-027/2010-027.pdf

TRAKLA2

Ari Korhonen
Helsinki University of Technology
P.O. Box 5400
02015 TKK
archie@cs.hut.fi

Juha Helminen
Helsinki University of Technology
P.O. Box 5400
02015 TKK
juha.helminen@cs.hut.fi

Ville Karavirta
Helsinki University of Technology
P.O. Box 5400
02015 TKK
vkaravir@cs.hut.fi

Otto Seppälä
Helsinki University of Technology
P.O. Box 5400
02015 TKK
oseppala@cs.hut.fi

ABSTRACT

TRAKLA2 is an online practicing environment for data structures and algorithms. The system includes visual algorithm simulation exercises, which promote the understanding of the logic and behaviour of several basic data structures and algorithms. One of the key features of the system is that the exercises are graded automatically and feedback for the learner is provided immediately.

The system has been used by many institutes worldwide. In addition, several studies conducted with the system have revealed that the learning results are similar to those obtained in closed labs if the tasks are the same. Thus, automatic assessment of visual algorithm simulation exercises provides a meaningful way to reduce the workload of grading the exercises while still maintaining good learning results.

Categories and Subject Descriptors

K.3.1 [**Computer Uses in Education**]: Computer-assisted instruction (CAI), Distance learning; K.3.2 [**Computer and Information Science Education**]: Computer science education—*data structures and algorithms*

Keywords

automatic assessment and feedback, visual algorithm simulation exercises, data structures and algorithms, trakla2

1. INTRODUCTION

Understanding the fundamental principles of data structures and algorithms is necessary for all programmers, and the ability to write efficient programs based on this knowledge is an important skill. Learning the concepts involved requires both theoretical knowledge and program comprehension skills. TRAKLA2 [11] provides a graphical environment for learners to practice and test their understanding of data structures and algorithms by solving short exercises.

Visualizations are a key part of explaining how data structures and algorithms work. They provide a way to abstract out implementation details and leave what is essential in order to understand the state of a data structure. Drawing such visualizations using pen and paper is an often used element of both homework and examinations.

In a typical pen-and-paper exercise the student might be asked to draw images of an AVL tree when a given series of operations is performed. The main drawback of this type of an exercise is the amount of time and effort spent on drawing. In many cases the changes to data structures are also very localized, thus most of the time is spent copying from the previous image.

TRAKLA2 transforms these pen-and-paper exercises into electronic form. The underlying idea of simulating the algorithm is the same. However, instead of being forced to repeatedly draw the structures, the user is given a set of corresponding visualizations that can be manipulated with a mouse. The idea is to perform the same transformations made by the algorithm being simulated. This allows the student to concentrate better on studying the algorithm.

Having more time on task is not the only benefit. Visual algorithm simulation exercises can be automatically graded and feedback given when the learner is still engaged with the problem. The grading is based on comparing the learner-made simulation sequence to a sequence produced by an actual algorithm implementation. The feedback for the learner is based on the number of correct steps in this simulation sequence. TRAKLA2 also randomizes the input data for algorithms, thus it discourages cheating and allows more opportunities for practicing.

The system is still evolving. Currently we have some 50 ready-made exercises for data structures and algorithms covering most of the topics studied on a typical CS2 course. In addition, we have a set of advanced exercises, for example, to cover courses such as spatial data algorithms. In their current form the exercises can effectively be used to complement or replace existing simulation-based homework. In addition, we are in the process of including new areas of

Koli Calling '09 October 29-November 1, 2009, Koli, Finland

interest and a variety of new kind of exercises for basic programming courses.

The rest of the paper is divided into the following sections. Section 2 describes the system, and in Section 3 we discuss related tools. Section 4 explains how the system has been adopted and disseminated. Section 5 briefly introduces some of the numerous evaluations done on the system. Finally, Section 6 gives some future directions.

2. DESCRIPTION OF THE TOOL

TRAKLA2 is an environment for supporting the learning of data structures and algorithms. It is based on the Matrix algorithm simulation framework [5] written in the Java programming language. The system provides automatically assessed visual algorithm simulation exercises that are meant to be accompanied by some other study material such as lectures, lecturer's notes, textbook, etc. Indeed, TRAKLA2 is primarily intended to be used as a practicing environment rather than a stand-alone learning environment. The current selection of exercises includes operations on binary trees, heaps, sorting algorithms, various dictionaries, hashing methods, and graph algorithms. See Table 1 for a complete list of the simulation exercises.

The algorithm simulation exercises are solved in the following manner. The student simulates the operations carried out in a given algorithm by manipulating data structure visualizations. No coding or typing of text is required, since all manipulation is carried out in terms of graphical user interface operations. The system records the sequence of operations, and finally allows the student to submit it to the server. On the server side, the student's sequence is compared to a sequence generated by a working implementation of the algorithm, and the student is given immediate feedback. There exists also a version that does not need any server connection in which case the grades are not stored anywhere. Thus, the client can also check the solution, which reduces the time it takes to give feedback. The grade of the student's solution is stored on the server, and the teacher can monitor students' points and other statistics. After receiving the feedback, the student can retry the exercise, if needed. The initial data for the exercise, i.e. the input for the algorithm, is randomly generated for each try.

The primary user interface for the exercises (see Figure 1) is a Java applet. The applet provides visualizations of data structures, push buttons for requesting Reset, Submit, and Model solution for the exercise as well as buttons for browsing one's own solution backwards and forwards. Simulation is carried out by drag-and-dropping data items (e.g., keys to be sorted by a sorting algorithm or records to be inserted into a binary search tree) or references (i.e., pointers in linked lists and trees) from one position to another. Some of the exercises also include push buttons to perform exercise-specific operations such as rotations in trees.

TRAKLA2 has been released as open source under the Apache License. The Matrix framework required by the simulation exercises has been released under the GNU General Public License (GPL).

3. RELATED TOOLS

At least two systems provide exercises similar to the ones in TRAKLA2, namely MA&DA [8] and PILOT [1]. Neither of the tools, however, enable an instructor to manage student accounts and points. Furthermore, the sets of available exercises are not even nearly as comprehensive as in TRAKLA2.

In addition, there are numerous systems that visualize algorithms and programs, and provide other kinds of interaction ranging from algorithm animations to pop-up questions. However, the number of such tools is too extensive to be listed here, thus we only give a couple of examples. JHAVÉ [12] and ViLLE [3] both include pop-up questions in which the student is expected to predict some aspect of the animation. JHAVÉ is targeted to algorithm visualizations and ViLLE for program visualizations. Both systems have recently been integrated with TRAKLA2, however. Moreover, JHAVÉ and the corresponding TRAKLA2 exercises have been embedded into tutorials that deal with the algorithms in question, thus providing additional learning material. The differentiating characteristic of TRAKLA2 exercises is the aim to activate the learner to trace an algorithm (learning by doing) instead of merely observing the visualizations. The most important feature in this learning process is the feedback received from the system, which is automatically generated. The feedback is immediate and speeds up the assessment cycle, which we believe promotes learning.

4. DISSEMINATION

The hosting of the system has previously been provided by the Helsinki University of Technology with funding from the Ministry of Education in Finland for the Network project on basic programming studies to promote best practices. It has been employed by nine (9) different institutions in Finland: Helsinki University of Technology, University of Turku, Tampere University of Technology, Helsinki Polytechnic - Stadia, Lappeenranta University of Technology, University of Helsinki, University of Kuopio, Åbo Akademi University, and one in the US: Indiana University East, Richmond, Indiana. In total, the system has been used on over 50 courses, by over 6500 students who have submitted over 380,000 solutions to the exercises.

The system has been installed by other universities as well. In addition, there is also a commercial provider that hosts TRAKLA2 internationally (in Europe, Asia, Africa and Australia) as part of its service[1].

5. EVALUATIONS OF THE TOOL

Over the years, many evaluation studies about TRAKLA2 and the simulation exercises have been carried out. Here, we present the most important results and introduce the various research questions addressed. For a more complete list of studies, see the list of publications on the TRAKLA2 research site.

In one of the first studies, we compared the learning results of students solving algorithm simulation exercises in instructed classroom sessions with students using a web-based learning environment [7]. The study concluded that there was no significant difference in the final exam results between the randomized student groups if the exercises were the same. Thus, some of the classroom activities can be replaced with this type of web-based material.

Several studies have focused on the use of resubmissions in the system. One of the key findings is that an encouraging grading policy combined with the option to resubmit

[1] `http://www.bythemark.com/`

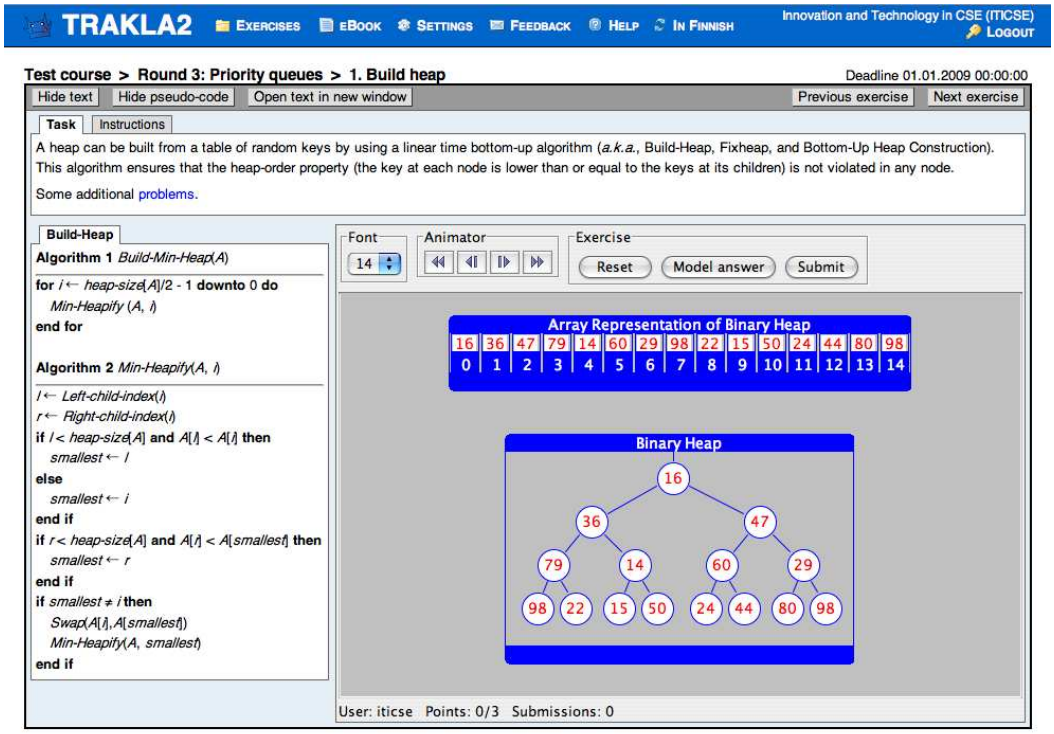

Figure 1: TRAKLA2 exercise view. The assignment instructions are located in the top area of the window together with some navigation buttons. The algorithm to be simulated is defined on the left and the data structures to be manipulated are on the right.

Table 1: List of available TRAKLA2 exercises.

Topic / Exercise	**Topic** / Exercise	**Topic** / Exercise
Basic algorithms	**Search trees**	**Algorithm Analysis**
Binary search	Binary Search Tree Search	Order of Growth
Interpolation search	Binary Search Tree Insertion	Running time of recursive algorithms
Preorder	Binary Search Tree Deletion	Running time of iterative algorithms
Preorder with stack	Faulty binary search tree	Asymptotic analysis
Inorder	Digital Search Tree	**Spatial Data Algorithms**
Postorder	Radix Search Tree	Point in Polygon
Levelorder	Single rotation	Point in Polygon with R-Tree
Postfix evaluation	Double rotation	Douglas-Peucker Line Simplification
Infix to Postfix	AVL-tree insertion	Closest pair of points
Dynamic programming	Red Black Tree	Point-Region Quadtree Insert
Sorting algorithms	Red Black Tree Coloring	R-Tree Insert
Quicksort	B-Tree	Polygon Traversal
Radix-exchange-sort	Trie	Line Sweep
Counting methods for sorting	**Hashing**	Visibility with Rotational Sweep
Insertion sort	Linear probing	Expanding Wave-method
Mergesort (iterative/recursive)	Quadratic probing	Voronoi Construction
Heapsort	Double hashing	Adding a point to TIN
Selection sort	Separate chaining	Polygon Skeleton
Priority queues	Rehashing	
Build heap	**Graph algorithms**	
Heap operations	BFS	
MinHeap Insert	DFS	
MinHeap Delete	Prim's algorithm	
	Dijkstra's algorithm	
	Dijkstra's algorithm with heap	

the solutions is an important factor in promoting students' learning [10]. In addition, the students solving more exercises get better grades. However, the article notes that in order to prevent the aimless trial-and-error method of problem solving, the number of resubmissions allowed per assignment should be carefully controlled. Fortunately, another study found that only a small number of students actually resubmit without thinking in between submissions [4].

In a more recent empirical study [9], collaborative learning on different Extended Engagement Taxonomy levels was examined with learning materials related to the binary heap that used visualizations on the different levels [6]. Pre- and post-tests were used as the test instruments in the experiment. In the study, statistically significant differences were found in favor of the students on the higher level of engagement, where also the TRAKLA2 exercises fall, in the total and pair average of the post-test scores.

In another study, students' simulation sequences recorded in TRAKLA2 were analyzed to infer existing student misconceptions of the build heap algorithm [13]. The results of this study suggest that misconceptions about how specific algorithms operate might be automatically recognizable from the simulation sequences. Such information would have high value for educators both for preventing and adressing misconceptions.

6. FUTURE WORK AND WEBSITE

In addition to the simulation exercises, other types of exercises and learning tools can be integrated into the system. Already, ViLLE system [3] provides exercises for the basic programming course (CS1) that can be incorporated into TRAKLA2. As mentioned, JHAVÉ [12] system has also been integrated into TRAKLA2. In the future, we are planning to include other similar tools such as Jype [2] to cover even more topics and courses. The idea is to avoid the need to learn to use many different systems by the learners, and thus focus on learning the content of the courses instead.

The project can be followed on a website that is located at `http://svg.cs.hut.fi/TRAKLA2/`. You can easily create a test account or utilize the stand-alone exercise applets without logging in. In addition, Koli Calling Conference maintains a website at `http://cs.joensuu.fi/kolistelut/tools/` that contains a brief overview of the system, a short video about the system, this paper, and some additional material.

7. ACKNOWLEDGMENTS

We thank the numerous people that have been involved in designing, implementing, testing, developing, and evaluating TRAKLA2 and its earlier versions during the past 18 years for their invaluable contributions.

This work was supported by the Academy of Finland under grant number 210947 as well as Ministry of Education, Finland.

8. REFERENCES

[1] S. Bridgeman, M. T. Goodrich, S. G. Kobourov, and R. Tamassia. PILOT: An interactive tool for learning and grading. In *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, pages 139–143. ACM Press, New York, 2000.

[2] J. Helminen. Jype – an education-oriented integrated program visualization, visual debugging, and programming exercise tool for python. Master's thesis, Department of Computer Science and Engineering, Helsinki University of Technology, March 2009.

[3] E. Kaila, T. Rajala, M.-J. Laakso, and T. Salakoski. Automatic assessment of program visualization exercises. In A. Pears and L. Malmi, editors, *Proceedings of the Eighth Koli Calling International Conference on Computing Education Research (Koli Calling 2008)*. Uppsala University, 2008.

[4] V. Karavirta, A. Korhonen, and L. Malmi. On the use of resubmissions in automatic assessment systems. *Computer Science Education*, 16(3):229 – 240, September 2006.

[5] A. Korhonen. *Visual Algorithm Simulation*. Doctoral dissertation (tech rep. no. tko-a40/03), Helsinki University of Technology, 2003.

[6] A. Korhonen, M.-J. Laakso, and N. Myller. How does algorithm visualization affect collaboration? video analysis of engagement and discussions. In J. Filipe and J. Cordeiro, editors, *Proceesing of the 5th International Conference on Web Information Systems and Technologies*, pages 479–488, WEBIST 2009, 23-26 March, Lisboa, Portugal, 2009. INSTICC — Institute for Systems and Technologies of Information, Control and Communication.

[7] A. Korhonen, L. Malmi, P. Myllyselkä, and P. Scheinin. Does it make a difference if students exercise on the web or in the classroom? In *Proceedings of The 7th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'02*, pages 121–124, Aarhus, Denmark, 2002. ACM Press, New York.

[8] M. Krebs, T. Lauer, T. Ottmann, and S. Trahasch. Student-built algorithm visualizations for assessment: flexible generation, feedback and grading. In *ITiCSE '05: Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, pages 281–285, New York, NY, USA, 2005. ACM Press.

[9] M.-J. Laakso, N. Myller, and A. Korhonen. Comparing learning performance of students using algorithm visualizations collaboratively on different engagement levels. *Journal of Educational Technology & Society*, 12(2):267–282, 2009.

[10] L. Malmi, V. Karavirta, A. Korhonen, and J. Nikander. Experiences on automatically assessed algorithm simulation exercises with different resubmission policies. *Journal of Educational Resources in Computing*, 5(3), September 2005.

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

[12] T. Naps. JHAVÉ: Supporting Algorithm Visualization. *Computer Graphics and Applications, IEEE*, 25(5):49–55, 2005.

[13] O. Seppälä, L. Malmi, and A. Korhonen. Observations on student misconceptions – a case study of the build-heap algorithm. *Computer Science Education*, 16(3):241–255, September 2006.