Informatics in Education, 2005, Vol. 4, No. 1, 49–68
© 2005 *Institute of Mathematics and Informatics, Vilnius*

Multi-Perspective Study of Novice Learners Adopting the Visual Algorithm Simulation Exercise System TRAKLA2

Mikko-Jussi LAAKSO, Tapio SALAKOSKI
Department of Information Technology, University of Turku
Lemminkäisenkatu 14 A, FIN-20520 Turku
e-mail: {milaak,tapio.salakoski}@it.utu.fi

Linda GRANDELL, Xuemei QIU
Åbo Akademi University
Department of Computer Science, Turku Centre for Computer Science
Lemminkäisenkatu 14 A, FIN-20520 Turku
e-mail: {linda.grandell, xuemei.qiu}@abo.fi

Ari KORHONEN, Lauri MALMI
Department of Computer Science and Engineering, Helsinki University of Technology
PL 5400, FIN-02015 HUT
e-mail: {archie,lma}@cs.hut.fi

Received: December 2004

Abstract. This paper presents results from three interrelated studies focusing on introducing TRAKLA2 to students taking courses on data structures and algorithms at the University of Turku and Åbo Akademi University in 2004. Using TRAKLA2 they got acquainted with a completely new system for solving exercises that provided them with automatic feedback and the possibility to resubmit their solutions. Besides comparing the students' learning results, a survey was made with 100 students on the changes in their attitudes towards web-based learning environments. In addition, a usability evaluation was conducted in a human-computer interaction laboratory.

Our results show that TRAKLA2 considerably increased the positive attitudes towards web-based learning. According to students' self-evaluations, the best learning results are achieved by combining traditional exercises with web-based ones. In addition, the numerical course statistics were clearly better than in 2003 when only pen-and-paper exercises in class were used. The results from the usability test were also very positive: no severe usability problems were revealed; in fact, the results indicate that the system is very easy to learn and user-friendly as a whole.

Key words: automatic assessment, feedback, computer science education, usability, evaluation.

1. Introduction

Automatic assessment (AA) tools for courses in computer science (CS) are being developed and gaining acceptance more and more widely in education at university level. The survey of the ITiCSE working group ’How shall we assess this’ in 2003 clearly indicated that the experience of using AA tools correlates with positive attitudes towards applying such methods more widely, also when assessing higher order skills (Carter *et al.*, 2003). The most common area in which AA is extensively used is assessing programming exercises (e.g., (Higgins *et al.*, 2002; Luck and Joy, 1999; Saikkonen *et al.*, 2001; Vihtonen and Ageenko, 2002)). Other applications include grading algorithm exercises (Bridgeman *et al.*, 2000; Hyvönen and Malmi, 1993; Korhonen and Malmi, 2000) and analyzing object-oriented designs and flowcharts (Higgins *et al.*, 2002).

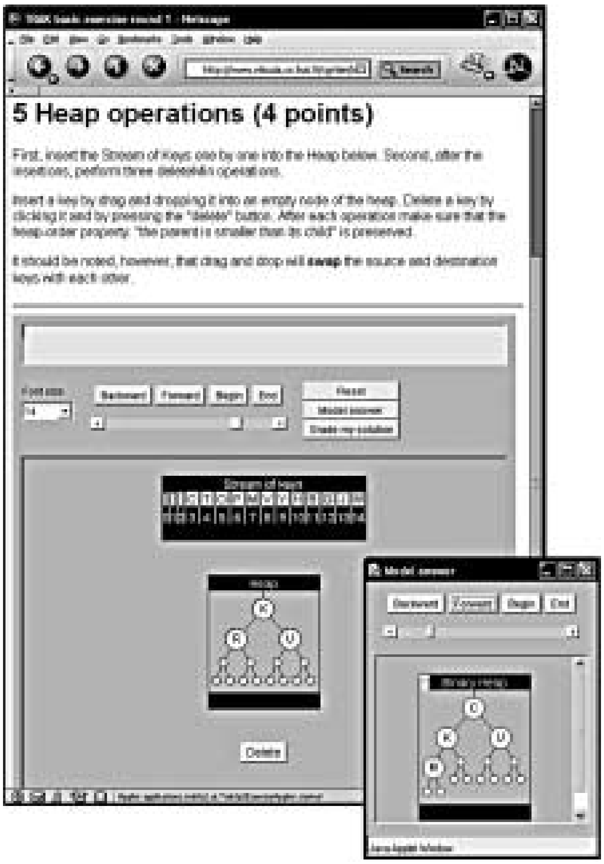
TRAKLA2 by Korhonen *et al.* (2003a,b) is a visual algorithm simulation exercise system that has been developed at Helsinki University of Technology (HUT). Students simulate how algorithms work on a conceptual level solving exercises using graphical manipulation of visualizations of data structures on the screen. The system provides automatic summative feedback on their work, and allows for resubmission of the solutions.

In 2004, TRAKLA2 was introduced at two universities in Turku, and in this paper we report the experiences from three interrelated studies focusing on using TRAKLA2: the first study compares the students’ learning results between using and not using TRAKLA2 in a course on data structures and algorithms (DSA). The second study consisted of a survey made with 100 students on the changes in their attitudes towards web-based learning environments. Finally, a usability evaluation was conducted in order to assess various usability aspects. This was the first time the usability of TRAKLA2 was studied by observing users interacting with the system in a human-computer interaction laboratory; the study can therefore be regarded as a pilot evaluation in this aspect. Before presenting the studies and the results in detail, we will start by giving an overview of the TRAKLA2 system. In the end of the paper, we will present the main conclusions and some suggestions for future work.

2. Overview of the TRAKLA2 System

TRAKLA2 is a system for automatically assessing *visual algorithm simulation exercises* (Korhonen *et al.*, 2003b). It is based on the Matrix algorithm visualization, animation, and simulation framework by Korhonen and Malmi (2002). TRAKLA2 distributes individually tailored tracing exercises to students and automatically assesses answers to the exercises. In visual algorithm simulation exercises, a learner directly manipulates the visual representation of the underlying data structures to which the algorithm is applied. The learner manipulates these real data structures through GUI operations with the purpose of performing the same changes on the data structures that the actual algorithm would do. The answer to an exercise is a sequence of discrete states of data structures resulting from application of the algorithm, and the task is to determine the correct operations that will cause the transitions between each two consecutive states.

Let us consider the exercise in Fig. 1. The learner has started to manipulate the visual representation of the Binary Heap data structure by invoking context-sensitive *drag-and-drop operations*. In the next step, for example, he or she can drag the key C from a *Stream*



5 Heap operations (4 points)

First, insert the Stream of Keys one by one into the Heap below. Second, after the insertions, perform three deleteMin operations.

Insert a key by drag and dropping it into an empty node of the heap. Delete a key by clicking it and by pressing the "delete" button. After each operation make sure that the heap order property "the parent is smaller than its child" is preserved.

It should be noted, however, that drag and drop will **swap** the source and destination keys with each other.

Fig. 1. TRAKLA2 applet page and the model solution window.

of keys into the left subtree of R in the binary heap. After that, the new key is sifted up via a *swap* with its parent until the parental dominance requirement is satisfied (the key at each node is smaller than or equal to the keys of its children). The swap operation is performed by dragging and dropping a key in the heap on top of another key. In addition, the exercise applet includes a push button for activating the Delete operation. The `Delete` button is applied in the second phase of the exercise to simulate the deleteMin operation. The student selects a node to be deleted and thereafter uses swap operations to heapify the tree again.

An exercise applet is initialized with proper *randomized input data*. The binary heap exercise, for example, is initialized with 15 alphabetic keys (Stream of keys) that do not contain duplicates. This means that the exercise can be initialized in more than 10^{19} different ways. The learner can *reset the exercise* by pressing the `Reset` button at any time. As a result the exercise is reinitialized with new random keys.

After attempting to solve the exercise, the learner can *review the answer* step by step using the `Backward` and `Forward` buttons. Moreover, the learner can *ask for feedback* on his or her solution by pressing the `Grade` button in which case the learner's answer is checked and immediate feedback is delivered. The feedback reports the number of correct steps out of the total number of required steps in the exercise. Finally, it is possible for the student to *submit the answer* to the course database using the `Submit` button. By default an answer to an exercise can be submitted unlimited times; however, a solution for a specific instance of an exercise with certain input data can be submitted only once. In order to resubmit a solution to the exercise, the learner has to reset the exercise and start over with new randomized input data. Thus, it is not possible to grade the same solution and improve it arbitrarily before submitting it.

A learner can also *examine the model solution* of an exercise. It is represented as an algorithm animation so that the execution of the algorithm is visualized step by step. In Fig. 1, the model solution window is opened in the front. The states of the model solution can be browsed using the `Backward` and `Forward` buttons. For obvious reasons, after opening the model solution for given input data, a student cannot submit a solution until the exercise has been reset and resolved with new random data.

Each TRAKLA2 exercise page (e.g., Fig. 1) consists of a description of the exercise, an interactive Java applet, and links to other pages that introduce the theory and examples of the algorithm in question. The current exercise set covers almost 30 assignments on basic data structures, sorting, searching, hashing, and graph algorithms. Appendix 7 lists the current exercises in TRAKLA2.

3. Study 1. Effect on Learning

3.1. *Background*

TRAKLA2 exercises were used for the first time in the basic DSA courses at HUT in spring 2003. The system was used in parallel with the older TRAKLA system so that a total of 14 TRAKLA2 exercises and 24 TRAKLA exercises were used in two courses[1]. In 2004, only TRAKLA2 was used and the total number of exercises was 26. During these two years more than 1000 students used the system.

In 2004, TRAKLA2 was also adopted in the DSA course at the University of Turku (UTU); a course attended by more than 100 students. Compared to the situation at HUT this was a major cultural change: at HUT automatically assessed algorithm simulation exercises have been used since 1991 using the older TRAKLA tool, and this type of exercises and the culture of using automatic assessment are thus well-established both among students and teachers. At UTU, however, no such exercises had been applied previously, except occasionally as pen-and-paper exercises without any automatic assessment.

In all these courses, both at HUT and UTU, the TRAKLA2 exercises constituted a compulsory part of the course, and grading points achieved from the exercises had an effect on the final grade of the courses, although in slightly different ways. At HUT,

[1] There were two versions of the course, one for CS majors and one for students of other engineering curricula.

TRAKLA2 exercises had an overall effect of 30% on the final course grade, whereas at UTU the TRAKLA2 exercises increased the points in the examination. The minimum requirements were the same at both universities: the students had to achieve at least 50% of the maximum points for the TRAKLA2 exercises.

3.2. *Settings*

The Course at UTU

The DSA course at UTU (DSA-UTU) in spring 2004 included 56 lecture hours, 10 classroom sessions (each 2 hours) and 22 TRAKLA2 exercises. Previous courses were given with 56 lecture hours and 13 classroom sessions (2 hours each). The classroom exercises consist of five single exercises, such as illustrating exercises, proof exercises, *etc.* TRAKLA2 exercises, however, are most effective to represent exercises in which the task is to illustrate how a specific algorithm works with given input data. Thus, the number of classroom exercises were cut down after TRAKLA2 was taken into use. In figures, the number of classroom exercises decreased from 65 to 50. Each TRAKLA2 exercise was given points from one to four. It was possible to get a maximum of 47 TRAKLA2 points in the DSA-UTU course. The TRAKLA2 exercises were divided into three rounds by synchronizing the exercises as different topics were covered in the course.

Grading and Requirements

It was possible to get 32 course points from the examination(s) and eight course points from the TRAKLA2 exercises. The conversion of TRAKLA2 points to course points was linear between the minimum requirements 50% (pass with zero course points) and 100% TRAKLA2 points (8 course points, i.e., 20% of the maximum).

There were two ways of passing the course: taking the final examination (0–32 course points) or taking two midterm examinations (both 0–16 course points). In either case, the students still had to fulfill the minimum requirements: i) do at least 20 of the 50 classroom exercises, ii) get at least 50% of the TRAKLA2 points (maximum 47 points), and iii) get at least 20 course points out of the total of 40.

In comparison with earlier DSA-UTU courses, TRAKLA2 exercises replaced one question in the examination or half a question in each midterm examination respectively. Traditionally one of the five questions in the examination has been of illustrative type, and this was the very question now replaced by TRAKLA2 exercises.

DSA-UTU course's final grading was on a scale from one to three with 0.25 steps. By getting 20 course points the student got the lowest grade (one). In addition, by doing 60% or 80% of the classroom exercises, any student could get an additional $+$ or $\frac{1}{2}$ to his/her grade, respectively (the students were still required to meet the minimum requirements of the course).

In the following, the data are compared to the data from the DSA-UTU course in spring 2003, when the course was given by the same lecturer and the classroom exercises were very similar to those in spring 2004.

3.3. *Results and Discussion*

As a whole, the TRAKLA2 system has worked well with surprisingly good results both at HUT and at UTU. In 2004, 30% of the students at HUT achieved the maximum number of points for the 26 exercises, and over 55% achieved 90% of the maximum. Only 15% of the students failed to get the required minimum of 50% of the points; in practice these were students who dropped out of the course at an early stage. At UTU the results were even better. The average number of points achieved was 7.34 points out of a maximum 8 points.

We will now present results derived from the course statistics, including the number of students failed/passed, average grades, attendances in classroom and TRAKLA2 exercises, and so on.

Table 1 shows statistics about student activity during classroom exercises from DSA-UTU courses in spring 2003 and spring 2004. In addition, students got an average of 7.34 course points from TRAKLA2 exercises, and 69.2 % of students did 100 % of TRAKLA2 exercises.

As we can see from the statistics, the spring 2004 students were more active not only in using TRAKLA2 but also in other parts of the course: in particular, the average performance in classroom exercises increased from 54.5% to 60.3%. There is also a significant statistical difference (χ^2-test, $p = 0.01$) between the two courses when considering the number of students in each class in Table 1. Moreover, a larger number of students received an additional $+$ / $\frac{1}{2}$ to their final grade in 2004 than in 2003. These observations confirm that the introduction of TRAKLA2 enhanced the students' motivation and work input in the DSA-UTU course.

Table 2 presents the basic statistics from DSA-UTU courses in 2003 and 2004. The statistics cover only results from the second midterm examination that was the first chance to pass the course. There was a major increase in the number of passed attendants. On the other hand, when looking at the average of course points (t-test, $p = 0.19$) and the average of final grades (χ^2-test, $p = 0.12$), there is no statistically significant difference between the two courses. Combining these two observations one can conclude that

Table 1

Student activity in classroom exercises

	Spring 2003	Spring 2004
Average % of classroom exercises done (only who did at least 40%)	54.5%	60.3%
Number of (#) attendants who did 0% - 40% of classroom exercises (failed)	76 (41%)	43 (26%)
# attendants who did 40% - 60% of classroom exercises (no bonus)	80 (43%)	79 (48%)
# attendants who did 60% - 80% and received + from classroom exercises to their final grade	18 (10%)	21 (13%)
# attendants who did 80% - 100% and received $\frac{1}{2}$ from classroom exercises to their final grade	12 (6%)	22 (13%)

Table 2

Course statistics of students taking the midterm examinations

	Spring 2003	Spring 2004
Number of (#) attendants in the course	186	165
# attendants who took the second midterm examination	58 (31%)	82 (50%)
# passed attendants (in midterm examinations only)	49 (26%)	81 (49%)
# attendants who took and failed the second midterm examination	9 (15.5%)	1 (1.2%)
Average course points	26.15	27.51
Average final grades	2.01	1.97

TRAKLA2 aided many students to get over the edge and pass the DSA-UTU course. Hence, TRAKLA2 seems to be truly useful for students who have difficulties in learning DSA by completing classroom exercises. The trend is very similar in examinations later in the course (actually, the average final grades were a little bit better in 2004 compared with 2003 if we look at the final statistics including midterm examinations and all the other examinations).

4. Study 2. Attitudes and Opinions

4.1. *Background*

Students' opinions about the system were gathered through a web-based survey at the end of the HUT course in 2003. 364 students answered. 80% of them gave an overall grade of 4 or 5 to the system on the scale 0–5, where 5 was the best grade. The system was almost unanimously considered to aid learning and to be easy to use. At UTU, free feedback about the system was well in line with the observations from the HUT survey. As a continuation on this survey we decided to also study the attitudes of the UTU students using questionnaires.

4.2. *Settings*

Three sets of questionnaires were filled out by the students during the DSA-UTU course. The first questionnaire was given at the beginning of the course, the second together with the first midterm examination (after the first round of TRAKLA2 exercises), and the third one with the second midterm examination (after the end of the course).

The purpose of the first questionnaire was to gather information about students' attitudes towards and experiences of web-based materials and tools in earlier courses. The students were also asked to state how well they thought web-based exercises could suite the DSA-UTU course (scale from 1 to 5, 5 being the best). The questionnaire also included a question on what kind of exercises the students would prefer to do in DSA-UTU courses as well as a question on how they assessed their own learning.

The second questionnaire had two main questions of yes/no type: the first question considered the contribution of TRAKLA2 to the students' learning of the course topics, and the second one covered the usability of TRAKLA2 and any potential usability problems. The students were also given the possibility to write free text comments related to these questions. In the third questionnaire, the questions from the first and second questionnaires were repeated. In addition, the students were asked for further comments and suggestions.

4.3. *Results and Discussion*

In addition, a different questionnaire was carried out which surveyed how students' attitudes towards on-line learning environments changed when they had used TRAKLA2: the results clearly indicated that the attitudes became more positive.

In the following, we present a more detailed analysis of the results of the survey on the UTU students' opinions and attitudes towards web-based learning. Moreover, the learning results based on students' self evaluation are presented.

There were 96 answers to the first questionnaire ('Start'), 103 to the second ('Mid'), and 81 to the third questionary ('End'). At the Start and End the students were asked about their opinion on the suitability of www-based exercises for learning DSA. The answer alternatives were *well* (5), *quite well* (4), *neutral* (3), *quite bad* (2), and *bad* (1). The Start average was quite high, 3.94, and the End average was even higher, 4.84. These results indicate that web-based exercises are perceived as very suitable for learning DSA. The large increase in the average during the course implies that web-based exercises were well accepted and appreciated even by students who had not demonstrated positive attitudes in advance.

The free text comments were also analyzed giving qualitative data. There was a number of answers in which students said that it is much more elegant to do this kind of illustrative exercises using TRAKLA2 instead of doing the same thing step-by-step using pen and paper. Moreover, many students mentioned that TRAKLA2 exercises concretized the actions and operations of an algorithm. It was also confirmed that the immediate feedback provided by TRAKLA2 helped the students find the point where they made a mistake, at the same time encouraging them to further deepen their understanding of the subject. This is also reflected by the course statistics.

In the Mid and End questionnaires, the students were asked how the TRAKLA2 exercises contributed to their learning. In the Mid, the question was formulated with only a yes/no-answer, whereby 94% of students answered that the TRAKLA2 exercises did aid their learning process. In the End, the students were asked to grade the contribution on a scale from 1 to 5 (5 being the best). The average of the answers was 4.10, and 84% of the students graded the contribution as 4 or 5, while there were only two answers below 3. This result corresponds well with previous results from studies at HUT.

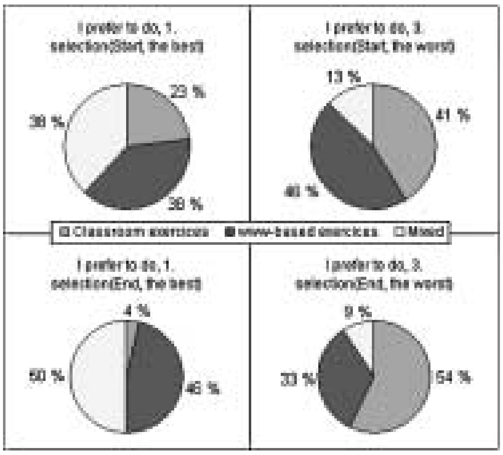


Fig. 2. I prefer to do.

We also asked the students to give their preference on three alternative ways of doing exercises: traditional classroom exercises, web-based exercises or a mix of these two approaches (see Fig. 2). In the same manner, the students were asked to assess the level of their learning (Fig. 3). The answers illustrate that the students' attitudes changed positively towards web-based exercises during the course. Even the results from the first questionnaire (Start) show that students prefer to do exercises by combining the traditional and web-based alternatives, and this opinion was strengthened during the course so that at the end, nearly three out of four students considered mixed exercises the best approach. Similar observations can be made about how the students assessed their own learning.

The mixed alternative is clearly the most appropriate approach to learn DSA. Fur-

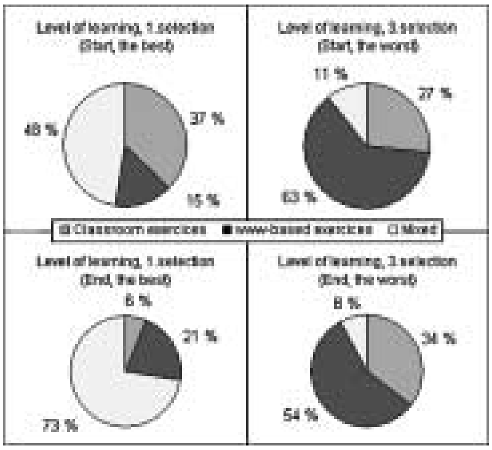


Fig. 3. The level of learning.

thermore, if the students were to choose between traditional and web-based exercises, they would prefer traditional over web-based ones due to the larger role of traditional exercises in facilitating learning. This is a very interesting result suggesting that although web-based exercises complement traditional ones done in class very well, the former can hardly replace the latter (in the current form).

5. Study 3. Usability Test

5.1. *Background*

Many studies have shown the effect of visualization software on learning outcomes in CS education (Hundhausen *et al.*, 2002; Naps *et al.*, 2003), but in order to be truly excellent, software systems should also be user-friendly. Whereas the usability of TRAKLA2 had been tested using questionnaires, this study was the first true usability evaluation including analysis of observations made in a usability laboratory.

When conducting usability tests, there are general guidelines and heuristics that can be used as a base for the evaluation. Naturally, all these general standards also apply to educational software. There are, however, specific issues that are especially important to consider when developing educational systems, such as focusing on learning efficiency, ensuring short response times, generating valuable feedback and motivating the learners. In this study we focused on the following usability aspects:

- learnability, intuitiveness and ease of use,
- usefulness and appropriateness in the curriculum,
- subjective satisfaction and motivation,
- efficiency for promoting learning.

5.2. *Settings*

The test was conducted with five students from the Department of Computer Science at Åbo Akademi University (ÅA) taking their first course on DSA, i.e., the kind of users the system has been designed for. The number of test participants might be considered low, but taking into account the limited resources available for this pilot evaluation such a small group sufficed. In addition, according to Nielsen (2000), five users are usually enough when conducting a usability test:

> As you add more and more users, you learn less and less because you will keep seeing the same things again and again. [...] After the fifth user, you are wasting your time by observing the same findings repeatedly but not learning much new.

To ensure that the students were on the same level concerning previous usage of TRAKLA2 and that their results could be considered comparable, none of the test participants had any prior experience in using the system.

Restrictions

We had many reasons for restricting the test to include only one of the exercises available in TRAKLA2: since the test participants were taking their first course on data structures, they were not yet familiar with all of the structures covered in TRAKLA2. In addition, the interface and functionality are quite similar for all exercises, wherefore testing one of these was assumed to have potential to reveal usability problems of the system as a whole. Finally, testing the entire system would be a laborious task, beyond the scope of this pilot test.

The exercise was decided upon in collaboration with the course lecturer in order to ensure that the chosen data structure and algorithm had been covered in the course up to the date of the test. The chosen exercise dealt with postorder traversal (POT) of binary trees.

Three-Part Study

The usability test consisted of three parts: observations, a pre- and post-test as well as a pre- and post-questionnaire.

The *observation sessions* took place one-by-one in a usability laboratory. Each individual session lasted approximately 30 minutes, of which the subjects spent 15 minutes interacting with the system completing the following scenarios: 1) Enroll, 2) Start the exercise on POT, 3) Use the system to solve exercises on POT and record the result from each attempt, and 4) Logout from the system.

The scenarios were put in logical order in order for the test to correspond with the authentic way of using the system. The two first scenarios can be regarded as rather trivial, giving the test participants a soft start to help them ’forget’ that they were in a test situation; anxiety and nervous feelings may affect the results. The third scenario was the main task, during which we were able to observe the participants using the system. The last scenario terminated the interaction, marking the end of the test session.

All observations were recorded as video and audio material for later analysis. In order to be consistent about the information given to the participants, the same test instructions were distributed on paper to each participant at the beginning of the observation session. The subjects were encouraged to think aloud throughout the session in order to reveal the thinking process while interacting with the system.

In order to evaluate the system’s effect on the students’ understanding of the specific data structure, *two pen-and-paper tests* were conducted; a pre-test one week before using the system and a post-test one week after. We decided not to give the post-test immediately after using the system in order to evaluate the system’s effect on the participants long-term learning.

In addition, *two questionnaires* were used: a brief pre-questionnaire was included in the pre-test in order to acquire background data about the participants. The post-questionnaire was given to the participants directly after they had finished interacting with the system and gathered information on their attitudes towards and experiences with TRAKLA2.

Table 3 illustrates how these three parts made it possible to analyze the usability aspects in focus.

Table 3

List of usability aspects in focus

Usability aspect	Observation	Pre/post-test	Post-questionnaire
Learnability & ease of use	x		x
Usefulness & appropriateness in curriculum			x
Subjective satisfaction & motivation	x		x
Effiency for promoting learning		x	

5.3. *Results and Discussion*

Observations

The data gathered from the observations were coded and analyzed using Noldus Observer software[2]. The observational material clearly indicates that the longer a subject worked with the system, the shorter the time required for solving an instance of the exercise. Whereas the mean time for solving the first exercise instance was 78.3 seconds, the corresponding time for solving the last one was 54.7 seconds. For one subject the difference between the longest and shortest duration for solving an exercise was as high as 110.0 seconds, the average difference being 60.7 seconds. The number of exercises solved during the 15 minutes was also high, the mean being 13.6 exercises.

The test participants spent most of the time during scenario three on solving exercises (80% in average), whereas 14% was spent on exploring, i.e., getting familiar with the interface, scrolling the screen and so on. The high percentage for solving actual tasks indicates that the subjects did not find it difficult to use the system in the intended way.

Unfortunately only a few of the test persons thought aloud actively, but those who did made good comments. The positive comments considered for instance the ease of dragging and dropping the keys and the possibility to change and review one's solution. There were no actual negative comments, only some recommendations. One of the subjects mentioned that the interface should be designed so that the exercise window would not require any scrollbars. This issue becomes clearer when the font size is increased. Another suggestion was to display the model answer window below the user's answer in order to facilitate comparisons and error detection. Automatically marking incorrect keys with another color was also expected to make it easier to find errors.

The analysis of the recorded material did not reveal any severe usability problems. When the users interacted with the system in an unexpected way, a 'Sorry'-dialogue was displayed on the top of the screen not giving any valuable information about what had gone wrong. The only option was to close the window, but since it was not modal some subjects did not close it; this made it pop up again every time the exercise was reset. In addition, one subject repeatedly pointed the mouse to the solution review area when attempting to reset the exercise; the majority of the subjects, however, did not have any problems with the navigation.

[2] The Observer is a system that can be used, e.g., to collect and analyse observational data. More information about the software can be found on http://www.noldus.com/site/doc200401012

Pre- and Post-tests

The pre- and post-tests both contained three assignments of corresponding nature. In addition to some POT problems, both tests included a part where the subjects were to describe how the algorithm on POT works. By this we were hoping to get information about how using the system had affected the subjects understanding of the algorithm and whether the system had helped the subjects create a mental model of how POT works (Ben-Ari, 2001), being able to explain it in own words could indicate that the subjects had reached a level of higher understanding. Unfortunately comparing the pre- and post-tests did not provide any valuable information on this point. The subjects' explanations were much shorter on the post- than on the pre-test. One can speculate that the algorithm seemed straightforward to the students after using TRAKLA2, and that they therefore did not feel any need for long explanations.

The two other assignments did, however, indicate that the participants performed better on the post-tests (see Table 4). On the first assignment the students were asked to give the order in which the keys in two binary trees would be visited using POT. On the pre-test the majority of the subjects got no points on this assignment, whereas almost everybody received a full score on the post-test. In the second assignment the subjects were asked to draw the binary tree corresponding to a given POT of keys. The difference between the post- and pre-test on this assignment was not as remarkable, but whereas three of the subjects did not receive any points at all on this assignment on the pre-test, everybody got at least two points on the post-test.

The score for each subject on the first two assignments on the pre- and post-test are given in Fig. 4. It is interesting to note that four out of five participants follow the same pattern, whereas one demonstrates an opposite behavior. Naturally, one cannot give any clear explanations for this phenomenon, but the common trend is still evident: for most participants the scores improved on the later test.

Post-questionnaire

In the post-questionnaire, the subjects were asked to grade TRAKLA2 on a five point Likert-scale (from 1 (disagree) to 5 (agree)) based on how they experienced the system. In Table 5 we give the median for the participants answers to questions that we found to be of importance in order to evaluate the first three usability aspects listed in Table 3.

As Table 5 shows, the system was regarded enjoyable and motivating. The data also imply that the participants were positive to using the system and most of them would

Table 4

Comparisons of the participants' performance on the pre- and post-tests

		Average	Std.dev
Assignment 1:	Pre-test	7	9.75
Give POT (max 20 p)	Post-test	18	4.47
Assignment 2:	Pre-test	4.2	5.76
Draw tree (max 11 p)	Post-test	4.4	3.71

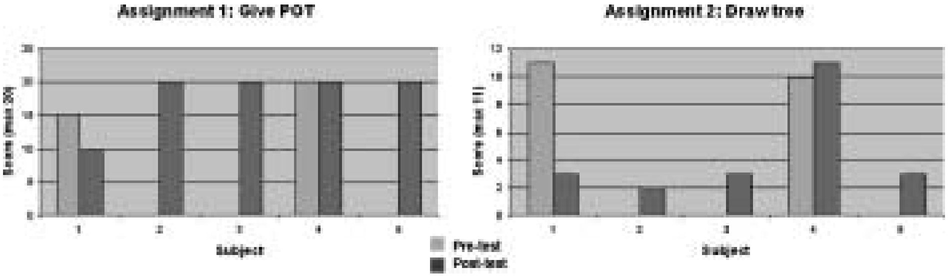


Fig. 4. Comparison of each subjects score on the two first assignments in the pre- and post-tests.

like to use it more extensively – in fact, we were asked to give them more information about the online version of the system. All except one of the participants reported that the system had helped them understand the algorithm on POT. The data indicates that the students managed to start using the system in a short time; learning how to use it was not considered difficult.

As a whole the usability evaluation has indicated that TRAKLA2 is easy to learn and use, thus fulfilling the first usability aspect: learnability and ease of use. The data from the post-questionnaire pointed out that the subjects would like to use TRAKLA2 in their studies and that they thought it would facilitate their learning (aspects 2 and 4: "Usefulness and appropriateness in the curriculum" and "Efficiency for promoting learning"). In addition, the results between the students varied more on the pre-test than on the post-test. This indicates that the system promoted the subjects' learning (aspect 4). Finally, the results from the post-questionnaire also imply that the subjects enjoyed working with the system and also wanted to have further usage. Such attitudes are in general very positive (aspect 3: "Subjective satisfaction and motivation").

Table 5

The median of the test participants' opinions after using the system (1 = disagree, 5 = agree)

	Median
Using the system was	
– frustrating	1
– enjoyable	4
– boring	2
– motivating	4
The system helped me understand the algorithm on POT	5
The system uses terms understandable and familiar to me	5
It took me a long time to start using the system	2
I would like to use the system in my studies	4
It would be easier to learn the topics in the course on data structures if I could use this system	4
Using the system was difficult	1
I solved the tasks quickly compared to the traditional pen and paper way	5

6. Conclusions

In this paper we have reported three studies, all focusing on the same system, TRAKLA2. It is interesting to see that the results from the studies support each other very well. In our opinion, testing a system in this way, from various perspectives, renders it possible to assess its true value, bringing out positive and negative aspects from different points of view.

Comparing the learning results from courses at UTU has shown that the introduction of TRAKLA2 has had a positive effect on students' learning. In addition, the first study pointed out that the TRAKLA2 system had a positive effect on students' behavior in other areas of the DSA-UTU course, for instance, the average student worked harder in order to learn the course topics. The possibility to get immediate feedback and to resubmit answers helped students to complete given exercises, thereby enhancing their learning. In addition to the positive trend in learning results, the number of passed attendants rose from 49 to 81; TRAKLA2 was extra helpful to less talented students in particular, supporting them to get over the edge and pass the course.

The vast questionnaire study at UTU has shown that students' attitudes strengthened positively towards web-based exercises. Moreover, the mixed alternative is far the most appropriate way to learn topics of courses on DSA, and it's well approved and preferred by students. Furthermore, the results suggest that web-based exercises constitute a good amendment to courses on DSA. However, there also appears to be a certain desire for more traditional exercises. Whether these students' expectations can be fulfilled by a future version of TRAKLA2 or similar web based tools, remains an interesting challenge. The results from the usability study indicate that it takes novices a very short time to learn to use TRAKLA2, and that even persons with no previous experience using the system could start working actively with it in only a few minutes.

7. Future Work

In our opinion, the studies have shown that TRAKLA2 is a usable system for enhancing learning of DSA. TRAKLA2 was well accepted and approved by students, and it will be used in forthcoming DSA courses in Turku.

At this time, the only type of exercises available in TRAKLA2 focuses on illustrating how a specific algorithm works on given input. Basically, this calls for tracing the execution of the algorithm. The system does not offer any support for constructive exercises, such as exercises in which a problem is described, example input and output values are given, and the task is to construct the algorithm. A key task of the future is to develop novel types of TRAKLA2 exercises as a collaboration between HUT, UTU and ÅA. In addition, now that a pilot evaluation in a usability laboratory has been conducted, TRAKLA2 could benefit from a larger-scale usability test, covering more exercises and letting the test persons interact more freely with the system.

A TRAKLA2 Exercises

Table 6

The visual algorithm simulation exercises in TRAKLA2 system. The column *name* describes the topic and the *description* characterizes the exercise. The roman numbers (i–iv) indicate the separate exercises and the number of sub-topics

Name	Description
Insertion into (i) Binary search tree, (ii) Digital search tree, and (iii) Radix search trie	The learner is to insert random keys into an initially empty search tree by dragging and dropping them into the correct positions.
Binary search tree deletion	The learner is to remove 4 keys from a binary search tree of 15 to 20 keys. Pointer operations are simulated by directly manipulating the edges that connect the nodes of the tree in the visualization.
Faulty Binary Search Tree	The learner is to show how to bring the following binary search tree in an inconsistent state: duplicates are allowed and inserted into the left branch of an equal key, but the deletion of a non-leaf node relabels the node as its successor.
AVL tree (i) insertion, (ii) single rotation, and (iii) double rotation	The learner is to (i) insert 13 random keys into an initially empty AVL-tree. The tree (i–iii) has to be balanced by rotations. The rotation exercises (ii–iii) require pointer manipulation, while the insertion exercise (i) provides push buttons to perform the proper rotation at the selected node.
Red-black-tree insertion	The learner is to insert 10 random keys into an initially empty Red-Black-tree. The color of the nodes must be updated and the tree must be balanced by rotations.
BuildHeap algorithm	The learner is to simulate the linear time buildheap algorithm on 15 random keys.
Binary heap insertion and delete min	The learner is to a) insert 15 random keys into a binary heap and b) perform three deleteMin operations while preserving the heap order property (see Fig. 1).
Sequential search: (i) Binary search, and (ii) Interpolation search	The task is to show which indices the algorithm visits in the given array of 30 keys by indicating the corresponding keys.
Tree traversal algorithms: (i) preorder, (ii) inorder, (iii) postorder, and (iv) level order	The learner is to show which keys in a tree the algorithm visits by indicating the visited keys in the required order.
Preorder tree traversal with stack	The learner is to simulate how the stack grows and shrinks during the execution of the preorder algorithm on a given binary tree.

To be continuated

Name	Description
Fundamental Graph algorithms: (i) Depth First Search, and (ii) Breadth First Search	The learner is to visit the nodes in the given graph in DFS, and BFS order.
Minimum spanning tree algorithms: Prim's algorithm	The learner is to add the edges into the minimum spanning tree in the order that Prim's algorithm would do.
Shortest path algorithms: Dijkstra's algorithm	The learner is to add the edged to the shortest paths tree in the order that Dijkstra's algorithm would do.
Open addressing methods for hash tables: (i) linear probing, (ii) quadratic probing, and (iii) double hashing	The learner is to hash a set of keys (10–17) into the hash table of size 19.
Sorting algorithms: (i) Quicksort, and (ii) Radix Exchange sort	The learner is to sort the target array using the given algorithm.

References

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, **20**(1), 45–73.

Bridgeman, S., M.T. Goodrich, S.G. Kobourov and R. Tamassia (2000). PILOT: An interactive tool for learning and grading. In *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*. ACM Press, New York, pp. 139–143. `citeseer.nj.nec.com/bridgeman00pilot.html`

Carter, J., J. English, K. Ala-Mutka, M. Dick, W. Fone, U. Fuller and J. Sheard (2003). ITICSE working group report: How shall we assess this? *SIGCSE Bulletin*, **35**(4), 107–123.

Higgins, C., P. Symeonidis and A. Tsintsifas (2002). The marking system for CourseMaster. In *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education*. ACM Press, pp. 46–50.

Hundhausen, C.D., S.A. Douglas and J.T. Stasko (2002). A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, **13**(3), 259–290.

Hyvönen, J., and L. Malmi (1993). TRAKLA – a system for teaching algorithms using email and a graphical editor. In *Proceedings of HYPERMEDIA in Vaasa*, pp. 141–147.

Korhonen, A., and L. Malmi (2000). Algorithm simulation with automatic assessment. In *Proceedings of The 5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*. ACM Press, New York, Helsinki, Finland, pp. 160–163.

Korhonen, A., and L. Malmi (2002). Matrix – Concept animation and algorithm simulation system. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM Press, New York, Trento, Italy, pp. 109–114.

Korhonen, A., L. Malmi and P. Silvasti (2003a). TRAKLA2: a framework for automatically assessed visual algorithm simulation exercises. In *Proceedings of Kolin Kolistelut/Koli Calling – Third Annual Baltic Conference on Computer Science Education*. Joensuu, Finland, pp. 48–56.

Korhonen, A., L. Malmi, P. Silvasti, J. Nikander, P. Tenhunen, P. Mård, H. Salonen and V. Karavirta (2003b). *TRAKLA2*. `http://www.cs.hut.fi/Research/TRAKLA2/` (27.9.2003).

Luck, M., and M. Joy (1999). A secure on-line submission system. *Software – Practice and Experience*, **29**(8), 721–740.

Naps, T.L., G. Rößling, J. Anderson, S. Cooper, W. Dann, R. Fleischer, B. Koldehofe, A. Korhonen, M. Kuittinen, C. Leska, L. Malmi, M. McNally, J. Rantakokko and R.J. Ross (2003). Evaluating the educational impact of visualization. *SIGCSE Bulletin*, **35**(4), 124–136.

Nielsen, J. (2000). *Why you only need to test with 5 users*. `http://www.useit.com/alertbox/20000319.html` (March 19, 2000)

Saikkonen, R., L. Malmi and A. Korhonen (2001). Fully automatic assessment of programming exercises. In *Proceedings of the 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*. ACM Press, New York, Canterbury, UK, pp. 133–136.

Vihtonen, E., and E. Ageenko (2002). Viope-computer supported environment for learning programming languages. In *The Proceedings of Int. Symposium on Technologies of Information and Communication in Education for Engineering and Industry* (*TICE2002*). Lyon, France, pp. 371–372.

M.-J. Laakso is a lecturer and a PhD student at Department of Information Technology, University of Turku. He received his MSc in computer science from University of Turku in 2003. His main interests are in the field of CS education research focusing on teaching programming to novices and visualization of algorithms.

T. Salakoski received his PhD degree in 1997 at University of Turku, Finland, and currently works as a professor of computer sience at Department of Information Technology, University of Turku. In addition to computer science education research, his scientific interests include developing intelligent data-analysis methods, resources, and tools for bioinformatics and linguistics.

L. Grandell is a PhD student at Turku Centre for Computer Science and the Department of CS at Åbo Akademi University. She received her MSc in computer science from Åbo Akademi University in 2003. Her main interests are in the field of CS education research focusing on teaching programming to novices both at secondary and university level.

X. Qiu is a researcher at the Turku Center for Computer Science, Institute for Advanced Management Systems Research, Åbo Akademi University. She received her MSc (econ) in information systems in 2003 from Åbo Akademi University. Her main research interests are interactive multi-agent systems in semantic Web environment and agent-enabled knowledge mobilization.

A. Korhonen is a researcher in the Helsinki University of Technology (HUT). He received his MSc (computer science) in 1997 and DSc (computer science) in 2003 in the HUT. He is acting as the scientific leader of the Software Visualization Group in the laboratory of Computer Science and Engineering. His research includes data structures and algorithms in software visualization, automatic assessment in computer science education, and various applications of computer aided learning environments.

L. Malmi is a professor of computer science in Helsinki University of Technology (HUT). He received his doctor of technology diploma in HUT in 1997. His main research area is computer science education including software visualization, automatic assessment, new educational methods, and evaluating how they improve learning.

Vizualios algoritmavimo mokymo simuliacinės sistemos TRAKLA2 multiperspektyvus tyrimas

Mikko-Jussi LAAKSO, Tapio SALAKOSKI, Linda GRANDELL, Xuemei QIU,
Ari KORHONEN, Lauri MALMI

Straipsnyje nagrinėjami rezultatai trijų tarpusavyje susijusių tyrimų, skirtų studentų supažindinimo su sistemos TRAKLA2 galimybėmis analizuoti. Atliekant tyrimus remtasi patirtimi, įgyta 2004 metais Turku ir Åbo Akademijos universitetuose dėstant duomenų struktūrų ir algoritmavimo kursus. Naudojant TRAKLA2 studentai buvo susipažindinami su visiškai nauja uždavinių sprendimo sistema, turinčia automatinį grįžtamąjį ryšį bei sudarančia galimybę peržiūrėti savo sprendimus. Norint palyginti besimokiusiųjų pasiektus rezultatus, buvo atlikta 100 studentų apklausa. Ja buvo siekiama išsiaiškinti, kaip keitėsi studentų požiūris į žiniatinkliu pagrįstą mokymosi aplinką. Tiriant kompiuterio ir žmogaus dialogą, sąsają, buvo atliktas ir sistemos praktiškumo įvertinimas.

Gauti rezultatai rodo, jog TRAKLA2 sistema padarė nemažą įtaką, kad atsirastų ir sustiprėtų teigiamas požiūris į žiniatinkliu pagrįstą mokymąsi. Atsižvelgiant į studentų atsakymus, galima teigti, kad geriausi rezultatai pasiekiami derinant tradicines ir žiniatinkliu pagrįstas užduotis. Be to, studentų pasiektų rezultatų statistika buvo kur kas geresnė nei 2003 metais, kai dėstant atitinkamus kursus buvo naudota vien „tušinuko ir popieriaus“ mokymo metodika. Sistemos praktiškumo testo rezultatai buvo taip pat teigiami: neaptikta jokių žymesnių panaudojamumo nesklandumų. Gauti rezultatai iš esmės rodo, jog sistema vartotojui nėra sudėtinga, ją nesunku suprasti, išmokti naudotis ja paprasta ir patogu.