Proceedings of the IASTED International Conference
WEB-BASED EDUCATION
February 16-18, 2004, Innsbruck, Austria

Collecting statistical data of the usage of a web based educational software

Panu Silvasti (psilvast@cs.hut.fi), Lauri Malmi (lma@cs.hut.fi), Petteri Torvinen (ptorvine@cc.hut.fi)

Department of Computer Science and Engineering
Helsinki University of Technology
PO Box 5400, 02015 HUT, Finland

ABSTRACT

TRAKLA2 is a web-based educational software for solving exercises on algorithms and data structures. The system has a cognitive tool for simulating algorithms visually and it can present model solutions of the exercises and assess students' solutions automatically. A method for collecting logging data of the usage of TRAKLA2 software and some results of analyzing the data gathered during the algorithms and data structures course at Helsinki University of Technology in spring term 2003 are presented. The study concludes that such analysis of the logging data can be an important method for evaluating the efficacy of the software in improving students' learning.

KEY WORDS

Educational software, Computer science education, Evaluation

1 Introduction

In this study a method for collecting statistical data of the usage of a web based TRAKLA2 educational software is presented. TRAKLA2 [5, 6] is an educational software tool for solving exercises on algorithms and data structures. TRAKLA2 has been developed in the Laboratory of Information Processing Science at Helsinki University of Technology (HUT) and it is currently used at the data structures and algorithms course, which has more than 600 students participating each year.

TRAKLA2 exercises are Java applets which can be embedded into a web-based learning environment. A TRAKLA2 exercise applet is a Learning Object [3] that is based on visualization of data structures and performing interactive operations on them. Currently there are over 20 exercises covering algorithms on basic structures, priority queues, dictionaries, and graphs implemented in the system. In the algorithmic exercise applet (see Figure 1) the learner modifies the visual representations of data structures simulating operations the actual algorithm would do. We call such a process visual algorithm simulation. In the particular exercise the learner simulates the binary search tree insertion algorithm by drag&dropping keys from the input stream into the tree structure. When the learner asks the system to grade his solution, the system assesses the answer and gives instant feedback, which reports the number of correct steps in the solution. In the model solution window the learner can browse an animation showing the discrete states of the binary tree structure as keys are inserted into the tree. Each time the exercise is initialized, the initial input data values are generated randomly. Moreover, after grading and watching the model solution, the exercise has to be reset with randomly generated initial values. There is no limit to how many times grading and feedback is allowed.

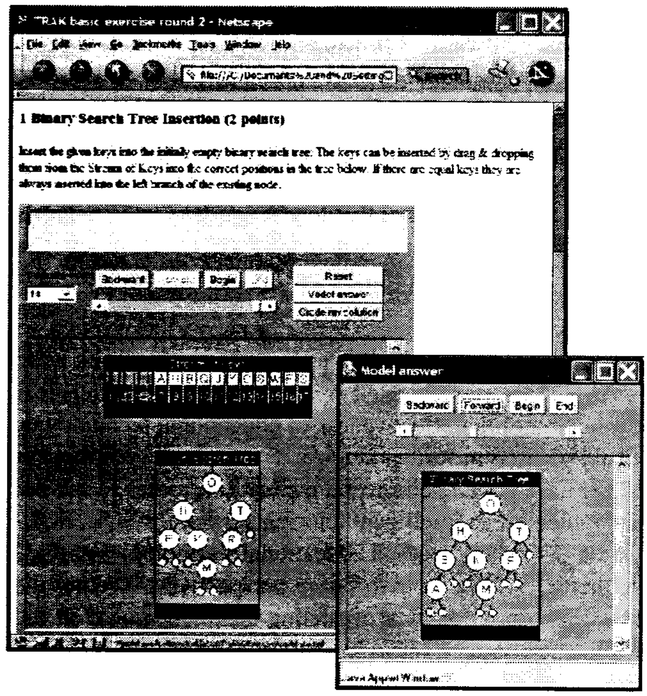

Figure 1. A TRAKLA2 exercise applet. The learner changes the visual representations of data structures by drag&drop operations. The learner can watch the model solution of the exercise, which is an animation of the algorithm. The learner can ask the system to grade his answer, in which case the system assesses the answer and gives instant feedback to the learner.

There are other cognitive tools applying algorithm simulation, such as PILOT [1] which is a tool for learning graph algorithms. However, TRAKLA2 has one important new feature: logging data of user's operations.

Students' personal learning processes and problem-solving methods affect how they use the TRAKLA2 exercise applet. They use different tools and sources of information in their process of learning. They can look up information from textbooks or web-sources; they can construct a mental image of the algorithm to be learned by discussing with other students or by making sketches with pen and paper. Eventually, each student must solve the TRAKLA2 exercises by using the user interface of the system. One can suppose that TRAKLA2 is not merely a user interface for constructing the answer of the exercise, but it is also a learning tool. Obviously several students use the feedback and model answer functions of TRAKLA2 applet for learning the algorithms and data structures in hand. In this study the logging data collected from the usage of the applet is studied in order to find out interesting details of students' problem solving in the system.

Evaluation is an important part of the development process of educational software. The goal of evaluation must be studying how the usage of educational software affects students' learning. In HUT the efficacy of automatic assessment and instant feedback in web-based algorithm simulation exercises has been studied. The study concluded that automatic feedback provided the same learning results than traditional closed laboratory exercises. When the number of students is very large, developing systems applying automatic assessment and feedback can be more cost effective than classroom teaching. [4]

The efficacy of automatic feedback provided by the grading functionality of TRAKLA2 is illustrated by showing the increase of the exercise grade in the relation to the number of times grading and automatic feedback were used. More sophisticated research frames for statistical evaluation of the effectiveness of educational technologies could be used in further studies. For example, Mitrovic et al [7] have conducted several evaluation studies in the development of an intelligent tutoring system for SQL, SQL-Tutor. They have studied educational technologies applied in the SQL-Tutor from different perspectives; whether the constraint-based student modeling techniques used in the SQL-Tutor supports learning, what is the impact of different levels of feedback in learning, and what is the effectiveness of pedagogical agents. They have used research frames where students were divided to different control groups and logging data of the use of SQL-Tutor was collected and analyzed statistically.

Understanding learners' actions and problem solving in educational software is also an important step for building adaptive systems. The next step for the development of TRAKLA2 web-based software will be the addition of adaptive operations, such as adaptive navigation support [2]).

2 Collecting data of user interface operations

The graphical user interface of a TRAKLA2 exercise applet is fairly simple. The basic operations: initialization of the exercise, opening the model answer, grading and submission are the same in each exercise. The solutions are defined in terms of algorithm simulation operations, i.e., editing visual representations of the data structures. The uniformity and simplicity of the user interface operations raises the idea that storage and analysis of the logging data collected from the user interface operations could be pretty simple. [8]

The technical solution for collecting data is based on creating Java objects corresponding to the user interface operations. The objects are sent to a server through Java RMI protocol. The server saves basic information of the log entries into a text file, which is easy to analyze. A log entry is saved each time the applet is initialized, the model answer window is opened or closed, and the exercise is graded or reset. A log entry contains a timestamp, course id, exercise id, student's user id, and the name of the operation performed. A log entry is also saved each time the user is idle over 60 seconds, and when the idle time ends to a user interface operation. At each grade operation a snapshot dump of students answer, which is a sequence of data structure states, is saved as a serialized Java object. [8]

TRAKLA2 exercises were used for the first time in the data structures and algorithms courses at HUT in spring 2003. There were two versions of the course, one for the CS majors and one for students of other engineering curricula. The TRAKLA2 exercises were a gradable and compulsory part of both courses. In total, some 600 students participated the courses and there were 14 different TRAKLA2 exercises, 8 for the CS majors' course and 6 for the other course. All the actions students performed with TRAKLA2 exercise applets were recorded into a database. Experiences from the use of data collection facility were positive. Data transfer from the applets to the server did not cause significant delays in using the applets. In some cases, though, strict firewall rules prohibited the communication between the applet and the server, in which case the exercise applet did not work.

3 Results

Figure 2 shows the differences of average operation counts in some exercises. The error bars denote the 95% confidentiality range. From the figure we can see that the average count of operations on the first four exercises (exercises on pre-, in-, post- and level-order traversal algorithms in binary trees) is less than in the exercises on binary heaps, and binary and interpolation search algorithms. It confirms the assumption that traversal algorithms are easier to learn than the other algorithms. The high value of initializations and using model solution in the interpolation search algorithm exercise fits well with this. Here one must remember that after opening the model answer the exercise is usually reset with new initial values, so each time the window is opened the model answer animation is different.

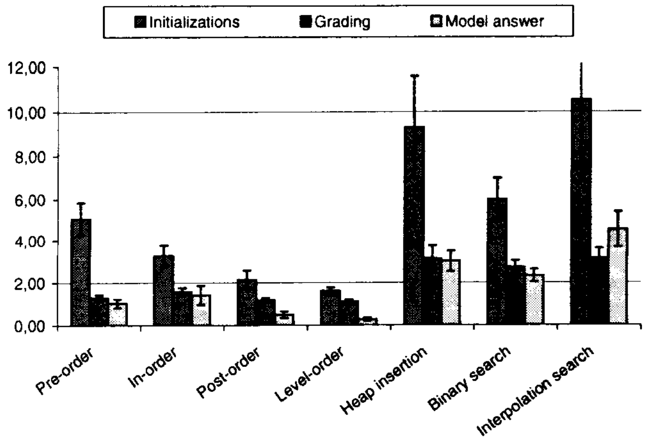

Figure 2. Average operation counts in different exercises. The sample sizes and error bars are included in the chart.

In Figure 3 the process of building the solution has been studied. The figure shows the development of exercise points each time when the exercise on building radix search tries was graded. Two bars in the series correspond to the count of students who finished the exercise at that attempt and their average points. 118 students passed the exercise at their first attempt getting 65% of maximum points. 87 students finished the exercise at their second attempt getting 81%, etc. Students who try five or more times before getting the exercise passed get maximum points from the exercise at their last attempt.

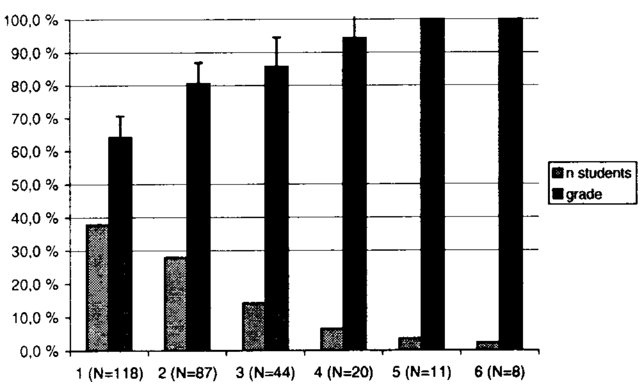

Figure 3. The development of points achieved from the exercise each time when the exercise was graded.

The model solution animation consists of a discrete sequence of visual snapshots of data structure states. The learner can browse the animation back and forth using the backward and forward buttons as depicted in Figure 1. Figure 4 shows the number of counts of students' visits in each state of the model solution animation of the heap insertion algorithm exercise, in which the learner has to insert 15 keys into an empty binary heap and perform 3 deleteMin operations. The figure confirms the assumption that the deleteMin-operation is more difficult to understand than the insert operation. The sample is taken from 73 students watching the model solution an average of 1.9 times.

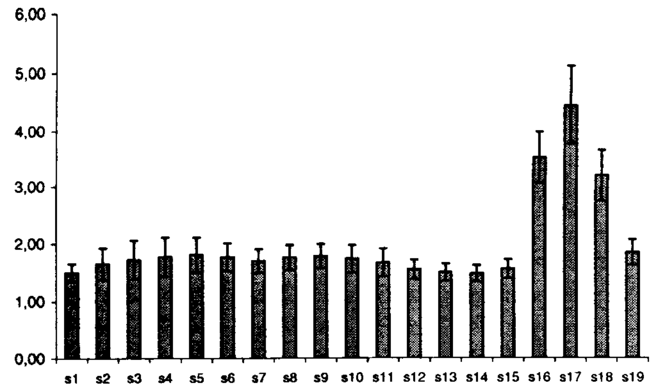

Figure 4. The counts of students' visits in each state of the model solution animation of the heap insertion algorithm exercise. The learner has to perform 15 insertions and 3 deleteMin-operations.

4 Conclusions

Collection and analysis of logging data of the usage of an educational software tool can help in understanding students' learning processes. This is very important in the perspective of evaluating educational software and developing adaptive systems. We have demonstrated that the data gathered from the TRAKLA2 system has provided useful information, which confirms several assumptions we have made on students' learning. Moreover, due to the large number of students we can get statistically reliable results.

This opens up interesting research questions for understanding student behavior better, and to improve the education methods in the future.

References

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

[2] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2–3):87–129, 1996.

[3] IEEE LTSC WG12. LOM: working draft document v6.4. learning object metadata working group of the ieee learning technology standards comittee. URL: `http://ltsc.ieee.org/files/wg12/`

LOM_1484_12_1_v1_Final_Draft.pdf (December 5, 2003), 2002.

[4] 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*, Aarhus, Denmark, 2002. ACM.

[5] A. Korhonen, L. Malmi, and P. Silvasti. TRAKLA2: a framework for automatically assessed visual algorithm simulation exercises. In *Proceedings of Kolin Kolistelut – Third Annual Baltic Conference on Computer ScienceEducation*, Joensuu, Finland, 2003.

[6] A. Korhonen, L. Malmi, P. Silvasti, J. Nikander, P. Tenhunen, P. Märd, H. Salonen, and V. Karavirta. TRAKLA2. http://www.cs.hut.fi/Research/TRAKLA2/.

[7] A. Mitrovic, B. Martin, and M. Mayo. Using evaluation to shape its design: Results and experiences with sql-tutor. *User Modeling and User-Adapted Interaction*, 12(2-3):243–279, 2002.

[8] P. Silvasti. Tilastollisen datan kerääminen algoritmisten harjoitustehtäväsovelmien käytöstä (Collecting statistical data of the usage of algorithmic exercise applets). Master's thesis, Department of Computer Science and Engineering, Helsinki University of Technology. (in Finnish), 2003.