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

Comparing Learning Performance of Students Using Algorithm Visualizations Collaboratively on Different Engagement Levels

Mikko-Jussi Laakso[1], Niko Myller[2] and Ari Korhonen[3]

[1]Department of Information Technology, University of Turku, 22014 Turun Yliopisto, Turku, Finland // milaak@utu.fi // Tel +358 2 333 8672 // Fax +358 2 333 8600

[2]Department of Computer Science and Statistics, University of Joensuu, P.O. Box 111, FI-80101 Joensuu, Joensuu, Finland // nmyller@cs.joensuu.fi // Tel +358 13 251 7929 // Fax +358 13 251 7955

[3]Department of Computer Science and Engineering, Helsinki University of Technology, P.O. Box 5400, FI-02015 TKK, Espoo, Finland // archie@cs.hut.fi // Tel +358 9 451 3387 // Fax +358 9 451 3293

ABSTRACT

In this paper, two emerging learning and teaching methods have been studied: collaboration in concert with algorithm visualization. When visualizations have been employed in collaborative learning, collaboration introduces new challenges for the visualization tools. In addition, new theories are needed to guide the development and research of the visualization tools for collaborative learning. We present an empirical study, in which learning materials containing visualizations on different Extended Engagement Taxonomy levels were compared, when students were collaboratively learning concepts related to binary heap. In addition, the students' activities during the controlled experimental study were also recorded utilizing a screen capturing software. Pre- and post-tests were used as the test instruments in the experiment. No statistically significant differences were found in the post-test between the randomized groups. However, screen capturing and voice recording revealed that despite the randomization and instructions given to the students, not all of the students performed on the engagement level, to which they were assigned. By regrouping the students based on the monitored behavior, statistically significant differences were found in the total and pair average of the post-test scores. This confirms some of the hypothesis presented in the (Extended) Engagement Taxonomy.

Keywords

Algorithm visualization, Algorithm simulation, collaborative learning, Engagement taxonomy

Introduction

Since its introduction, it has been hoped that Algorithm Visualization (AV) would solve problems related to learning of data structures and algorithms. However, empirical evaluations have yielded mixed results when determining the usefulness of such visualizations as teaching and learning aids over traditional methods (see the meta-analysis of the research on AV by Hundhausen et al. (2002)). Thus, researchers have sought explanations for the mixed results as well as better grounds to justify the use of visualizations in teaching. Hundhausen et al. (2002) concluded that the activities performed by the students are more important than the content of the visualization. This has led to the analysis of different engagement levels Naps et al. (2002) by ITiCSE Working Group that proposed *Engagement Taxonomy* (ET) to describe the various types of activities that students perform with visualizations and their effect on learning and Myller et al. (in press) have developed it further into *Extended Engagement Taxonomy* (EET).

Collaboration has become accepted and popular in Computer Science education. A good example is the benefits of pair programming (Nagappan et al., 2003; Williams et al., 2000; McDowell et al., 2003). Whilst visualizations are employed in collaborative learning, collaboration introduces new challenges for the visualization tools. For example, the exchange of experiences and ideas, and coordination of the joint work are needed when students are not working individually anymore (Suthers and Hundhausen, 2003). Furthermore, visualizations can provide a shared external memory that can initiate negotiations of meanings and act as a reference point when ideas are explained or misunderstandings are resolved (Suthers and Hundhausen, 2003). This implies that also new theories are needed to guide the development and research of the visualization tools for collaborative learning.

In this paper, the applicability of EET in collaborative use of visualizations has been studied. We test the impact of EET levels on the performance when visualizations are used in collaboration. We present an empirical study, in which learning materials containing visualizations on different EET levels were compared when student pairs were collaboratively learning concepts related to binary heap. The pairs had a mutual task to read through a tutorial including visualizations and answer questions related to the topic. Although, statistically significant differences were

not detected in a previous study, the results indicated that the engagement level of the visualizations has an effect on the performance when students are working in pairs (Myller et al., 2007). Thus, we replicated that study in a different institution, and improved the settings in such a way that the detection of the statistically significant differences would be possible. In this paper, we report the results from the replication study conducted at the Helsinki University of Technology in which two groups of students were randomized to the computer lab sessions. Each session was randomly assigned to an EET level, either *changing* or *controlled viewing* (in the rest of the paper this can be also shortened to viewing when we are discussing about the groups), with the limitation that parallel sessions belonged to different conditions.

During the analysis of the screen and voice recordings collected in the study, it was detected that despite the randomization and instructions given to the students, not all of the students performed their learning on the expected EET level. This meant that although the tool allowed students to learn on a higher EET level, some of the students choose not to do so, but worked on a lower engagement level. Fortunately, the screen capturing and voice recording done during the students' learning process provided us a tool for noticing this and taking it into account in the analysis. Thus, in addition to the results from the study, we learned an important methodological lesson as well. Screen capturing and voice recording should be a standard procedure, because otherwise we cannot know for sure if the participants really do what we expect them to do.

In Chapter 2, we describe the relevant literature related to the engagement taxonomy and similar theories. In addition, we give an overview of the learning tool used in the experiments. Chapter 3 describes the research setting, i.e., the used pre- and post-tests, subjects, materials, and procedures. In Chapter 4, we report on the results. Finally, in Chapters 5 and 6, we make conclusions and highlight some future directions.

Previous Research

Visualizations and Engagement

As an attempt to describe the mixed results of previous research in AV usage (cf. (Hundhausen et al., 2002)) in learning and teaching of algorithms and data structures, Engagement Taxonomy (ET) was introduced by Naps et al. (2002). The central idea of the taxonomy is that the higher the engagement between the learner and the visualization, the higher the positive effects on learning outcomes. ET consists of six levels of engagement between the user and the visualization:

No viewing	There is no visualization to be viewed.
Viewing	The visualization is only looked at without any interaction.
Responding	Visualization is accompanied with questions, which are related to the content of the visualization.
Changing	Modification of the visualization is allowed, for example, by varying the input data set or algorithm simulation.
Constructing	Visualization of program or algorithm is created.
Presenting	Visualizations are presented to others for feedback and discussions.

ET has been used in the development of AV tools and several studies have utilized the framework and provided further support for it (see, e.g., Grissom et al. (2003); Naps and Grissom (2002)). However, the time to study the materials on different ET levels has commonly been an uncontrolled variable in the studies, meaning that students have had freedom to use as little or as much time as they wanted to. Thus, those students who have been studying with visualizations that are on the higher ET level have spent more time on the task. This, in turn, makes it questionable if the reason for better performance in the post-test is due to the additional time spent on studying or the higher ET level of the materials. In the experiment, which is presented in this paper, we controlled the time so that all the students needed to spend exactly the same amount of time on learning the topic.

There are also other studies which have shown that visualizations improve learning, without actually utilizing the ET framework in the design of the study (Ben-Bassat Levy et al., 2003). In addition to this, research in educational psychology and multimedia learning had also had similar results (Evans and Gibbons, 2006).

Myller et al. (in press) have proposed an extension to the ET called the *Extended Engagement Taxonomy* (EET). The idea of this extension is to let the designers and researchers of visualizations to use finer granularity of engagement levels in their tools and experimental designs. They provide the following engagement levels to be used together with the original ones: *controlled viewing*, *providing input*, *modification*, and *reviewing*. In this study, we will utilize the controlled viewing level in order to make a difference between the visualizations that can only be viewed by the student (EET level: *viewing*, e.g., static visualizations or animations with only a playing option) compared to those which can be controlled (EET level: *controlled viewing*, e.g., animations with VCR-like controls in order to step and play the animation both forwards and backwards).

Visualizations and Collaboration

From a more general perspective, there are studies that analyze the use of visualizations in collaboration. For instance, Suthers and Hundhausen (2003) have performed research in the area of scientific inquiry. They compared the effects of different representations (i.e., matrix, graph, and text) when students were collecting and analyzing data, hypotheses and their evidential relations. Their research showed that the form of the visualization and what kinds of interactions it drives have an effect on the collaboration process by making certain data and their relations more explicit or implicit.

Roschelle (1996) studied pairs of students using the learning environment of Newtonian physics and analyzed their learning outcomes as well as the process that led to those outcomes. During the study, it was recognized that learning tools and especially visualizations used in collaboration should focus more on supporting communication rather than presenting the underlying model as accurately as possible. Furthermore, Roschelle (1996) tells as the last lesson in his paper that, "one should design activities, which actively engage students in doing and encountering meaningful experiential feedback as a consequence of their actions". Scaife and Rogers (1996) also identified the analysis of the interactions between external presentation and its users as a key research area for the future. All these points of view seem to support the applicability of ET/EET in the context of collaborative learning.

Although several AV tools have been developed and empirical studies carried out, the collaborative use of AV tools is researched very little. Myller et al. (in press) have studied the applicability of EET to describe differences in the learning process when visualizations are used during collaborative learning. They pointed out that when students were using visualizations on lower EET levels the interaction/engagement between students also dropped, meaning that students communicated and collaborated more when they were using materials on higher EET levels.

The work of Hundhausen (2002) is related to the collaborative aspects of AV construction and presentation. This work led into the development of a visualization tool, ALVIS, which supports construction and presentation of AVs in small groups (Hundhausen and Brown, 2008). Their results also indicate that ET is applicable in the context of collaborative learning, although it is not directly tested. Furthermore, Hundhausen (2005) has proposed a communicative dimensions framework in order to analyze the aspects of visualizations that affect communication between end-users. Hübscher-Younger and Narayanan (2003) developed a web-based system that allows students to post their own algorithm representations (e.g., text, pictures, animation, or multimedia) and discuss them on the web. The research concluded that the students who actively participated in this activity achieved higher grades than the passive students who might have only viewed and commented on others' presentations.

Other Algorithm Visualization Studies on Heap Data Structures

Stasko et al. (1993) utilized algorithm animations focusing on a pairing heap that was implemented as a binary tree. The results were disappointing: the animation group outperformed the control group but the differences were not high even on absolute scale, and the differences were not statistically significant. Moreover, they noted that using animations did not grant obvious learning benefits and they believe that algorithm animations benefit advanced students more than "novice students".

In 1996, Byrne et al. (1996) conducted algorithm animation research on binomial heap. The results were not statistical significant, either, and their findings supported the view that the benefits of animations are not that obvious, and careful task analysis is essential to determine in which situations animation can be helpful. Also Kehoe

et al. (2001) studied the learning of binomial heap through animations in open lab sessions. They hypothesized that animations make complex algorithms more accessible and less intimidating and enhance students' motivation, interaction and learning. Their study, however, was inconclusive (they made hypotheses), and further empirical studies were suggested.

There are some differences between these studies and ours. Our students were novices with little or no previous knowledge on the topic, but they were not novices in using the visualization tool but had previous knowledge on how to use the tool and how to make sense of its visualization. However, students needed to study in our experiment concepts related to binary heap, which might be easier to understand and more accessible for novices compared to the pairing heap or the binomial heap. Furthermore, we used fixed time limits for the learning session meaning that all students needed to use exactly the same time to learn the topic, and we monitored their learning process in order to detect how they were learning.

TRAKLA2 Overview

TRAKLA2 is a practicing environment for *visual algorithm simulation exercises* (Korhonen et al., 2004) that can be assessed automatically. The system distributes individually tailored tracing exercises to students and provides feedback about students' solutions automatically. In visual algorithm simulation exercises, a student directly manipulates the visual representation of the underlying data structures (i.e., a student acts on the EET level *changing*). Thus, the student manipulates real data structures through GUI operations with the purpose of performing the same changes on the data structures the actual algorithm would do. An answer to an exercise is a sequence of discrete states of data structures, and the task is to perform the correct operations that will cause the transitions between each of the two consecutive states.

Each TRAKLA2 exercise page consists of a description of the exercise with links to other pages that introduce the theory and examples of the algorithm in question, instructions on how to interact with the GUI, code window, and an interactive Java applet. The current exercise set consists of over 40 assignments on basic data structures, sorting algorithms, search trees, hashing methods, and graph algorithms.

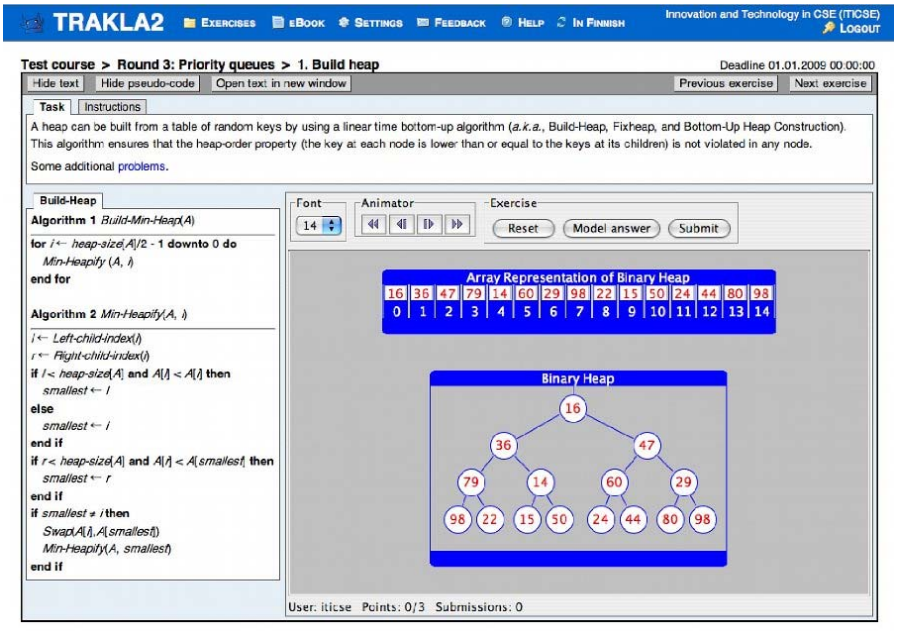

Figure 1: TRAKLA2 exercise page. The student acts in EET level *changing* by solving the exercise in terms of swapping the data elements in the data structure(s)

Let us consider the exercise in Figure 1. The student is supposed to manipulate the visual representation(s) of the Binary Heap data structure by invoking context-sensitive *drag-and-drop operations*. The idea is to simulate the

linear time BuildHeap algorithm. The manipulation can be done in either of the representations shown in the figure (i.e. the array or the binary tree representation). A key can be sifted up in terms of *swap operations* with its parent until the heap property is satisfied (the key at each node is smaller than or equal to the keys of its children). A single swap operation is performed by dragging and dropping a key in the heap on top of another key.

An exercise applet is initialized with *randomized input data*. The BuildHeap exercise, for example, is initialized with 15 numeric keys that correspond to the priority values. The student can *reset the exercise* by pressing the *Reset* button at any time. As a result, the exercise is reinitialized with new random keys. When attempting to solve the exercise, the student can *review the answer* step by step using the *Animator* panel. Moreover, the student can *Submit* the answer in which case the answer is assessed and immediate feedback is delivered. The feedback reports the number of correct steps out of the total number of steps in the exercise. This kind of automatic assessment is possible due to the fact that, again, the student is manipulating real data structures through the GUI. Thus, it is possible to *implement* the same algorithm the student is simulating, and execute it so that the algorithm manipulates the same data structures, but different instances, as the student just did. The assessment is based on comparison between these two different instances of data structures with each other.

An exercise can be submitted an unlimited number of times. However, a solution for a single instance of an exercise with certain input data can be submitted only once. In order to resubmit a solution to the exercise, the student has to reset the exercise and start over with new randomized input data. A student can also review a *Model answer* for each attempt. It is represented in a separate window as an algorithm animation accompanied with a pseudo code animation so that the execution of the algorithm is visualized step by step. The states of the model solution can be browsed back and forth using a similar animator panel as in the exercise. For obvious reasons — after opening the model solution — the student cannot submit a solution until the exercise has been reset and resolved with new random data.

TRAKLA2 visual algorith simulations and their instant feedback and model answer capabilities can also help students to collaborate with each other by providing shared external imagery and memory that can be processed together. Furthermore, they can increase the awareness of the students on each others abilities and knowledge (Collazos et al., 2007).

Previous Studies on TRAKLA2

In 2001, the first intervention study Korhonen et al. (2002) with three randomized groups A, B, and C ($N_A = 372$, $N_B = 77$, $N_C = 101$) was performed. Students' behavior was monitored over the second year course in data structures and algorithms (DSA) lasting twelve weeks. The examination results of students using the TRAKLA learning environment (predecessor of TRAKLA2) were compared with those in the traditional classroom sessions. The results showed that, if the exercises are the same, there is no significant difference in the final examination results between students exercising on the web (group A) or in the classroom (group B). In addition, the commitment to the course (low drop-out rates), is almost equal in both versions of the course. However, if the exercises are more challenging (group C), there is a significant difference in the examination results, but the drop-out rate is significantly higher as well.

Laakso et al. (2005a) reported on another whole semester study, in which TRAKLA2 was introduced at the University of Turku. The students' learning results were compared between students, who used or did not use TRAKLA2, during a course on DSA. In addition, a survey-data ($N = 100$) was collected on the changes in students' attitudes towards web-based learning environments. The results showed that TRAKLA2 considerably increased the positive attitudes towards web-based learning. According to students' self-evaluations, the best learning results were achieved by combining traditional and web-based exercises. In addition, the overall student performance was clearly better than in 2003 when only in class pen-and-paper exercises were used.

In 2005, the 2001 and 2004 studies were repeated at the Helsinki University of Technology (HUT) and at the University of Turku (UTU) during the spring semester (Laakso et al., 2005b). The students ($N = 133 + 134$) were divided into two randomized exercise groups in both universities. The first group started their exercises on the web with the TRAKLA2 learning environment while the second group did their exercises in classroom sessions. In order to prevent the high drop-out rates (see, group C in 2001), however, the same learning experience were provided for

all the students. At the midpoint of the course, the treatment for the students was changed. The first group continued in the class room and the second group on the web. Moreover, the same attitude survey, which carried out at UTU in 2004, was administered in both of the aforementioned universities.

The study concluded that it is good to introduce easy and guided exercises at the very beginning of the course. In addition to this, there is an emerging need for both web-based and classroom exercises. The recommended way to introduce the web-based exercises in DSA courses is by combining these two approaches. There is a set of exercises that are more suitable to be solved and automatically assessed on the web while the rest of the exercises are more suitable for traditional classroom sessions. More detailed information about this repetition study can be found in Laakso et al. (2005b).

The above studies were whole semester studies, in which the focus was on students' overall performance and drop-out rates. The difference between the treatments were in learning settings: the control groups were in classroom while the treatment groups were on the web. However, the learning objectives were the same for all groups, i.e., the exercises were algorithm simulation exercises. In addition, we studied the students' attitudes towards web based learning environments.

In contrast to the above studies, Myller et al. (2007) conducted an experimental study focusing on engagement taxonomy in fall 2006 at University of Turku. In the study, the learning outcomes of the students, who learned in collaboration by using visualization on different engagement levels were compared. There were 52 students in the treatment group (EET level: changing) and 53 students in the control group (EET level: controlled viewing), which sums up to 105 participants. The setup was a pre-test, treatment, post-test design. The post-test included the same questions as the pre-test, and additionally more difficult questions in order to see if the differences were apparent in them. The results indicated that the level of engagement had an effect on students' learning results in favor of the treatment group, although the differences were not statistically significant. Especially students without previous knowledge seemed to learn more from using visualizations on higher engagement level. In this paper, we report on a replication of this study with minor changes in order to repair the flaws in the design of the pre-test and post-test as reported by Myller et al. (2007).

Experimental Setup

To summarize the previous sections, the collaborative use of AV tools has been studied only little, yet the need for this kind of research emerges from the increasing use of visualization tools in collaborative learning. We hypothesize that the EET framework can be used to predict performance differences when visualizations are used in collaboration. Previous research supports this view and our hypothesis is based on the previous research on TRAKLA2 and formulated as follows: Students using visualizations collaboratively on EET-level *changing* (i.e. in pairs) perform better compared to students using only visualization on EET-level *controlled viewing* (again in pairs).

In order to test our hypothesis, we carried out an experiment in which we compared the learning outcomes of students who were collaboratively using visualizations which were on different EET levels. Participants were (mostly first year) Computer Science major students on a data structures and algorithms course at the Helsinki University of Technology. We utilized TRAKLA2 (Korhonen et al., 2004) in order to provide students with algorithm simulation exercises that act on the EET level *changing* (treatment group). However, the students did not have the option to reset the exercise to obtain a new similar exercise with new input data, but they had to work with a fixed input data for each exercise during the whole session. The animations that the students used in *controlled viewing* condition (control group) were similar to those used in model answers provided by the TRAKLA2 system.

Quantitative results were analyzed with one-tailed t-test, ANOVA and χ^2 -test depending on the nature of the data. We used the Bonferroni correction when applicable. The justification for using one-tailed t-test is based on the formulation of our hypothesis, which predicts that students using visualizations on EET-level *changing* perform better than students using visualization on EET-level *controlled viewing*. The hypothesis is based on the previous research in which it was found that student groups using visualizations on EET-level *changing* consistently performed better than student groups using visualization on EET-level *viewing* or *controlled viewing* although differences were not statistically significant (Myller et al., 2007).

Method 1: Experimental Study

The study was a between-subject design with pre-test and post-test (dependent variable). We had one between-subject factor (independent variable): the highest available EET level of the visualizations in the learning materials, namely *controlled viewing* or *changing*. The unit of analysis was either a student or a pair of students depending on the measure. Each student answered the pre- and post-test individually, but all the observational data collected during the pair learning is not individual but the same for the pair. Moreover, we also report the average performance of the pair in the post-test and use it in the analysis.

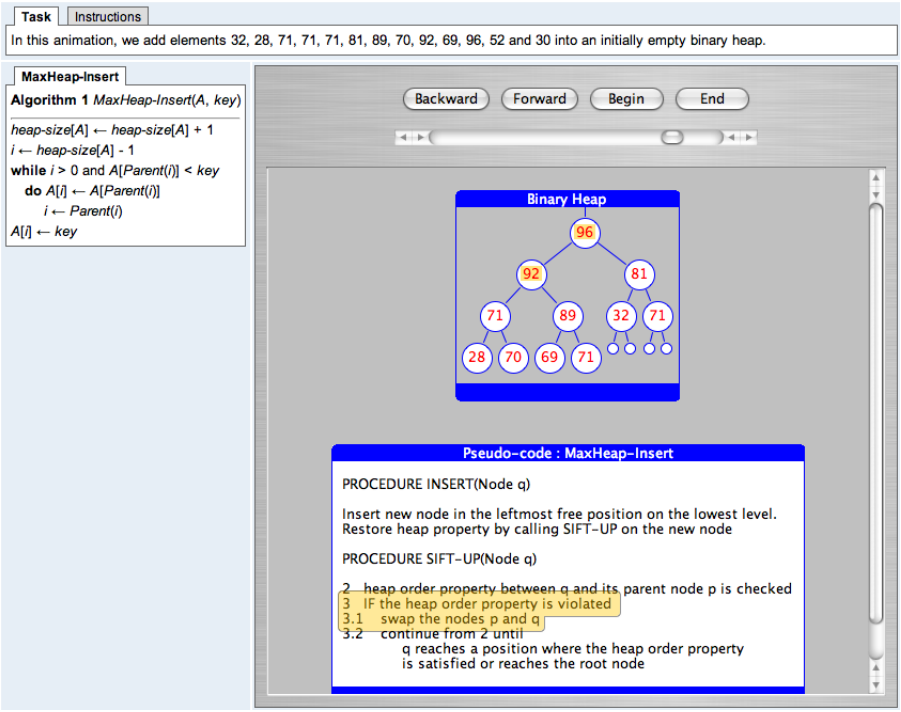

Figure 2: Binary heap insert animation in the tutorial. The student acts on EET level *controlled viewing*. The user has VCR like buttons (Backward, Forward, Begin, End) to interact with the animation

The learning materials contained textual materials that were the same for both conditions. In the *changing* condition, textual materials were accompanied with TRAKLA2 (Korhonen et al., 2004) algorithm simulation *exercises* related to the binary heap (see Figure 1). Student pairs in the *controlled viewing* condition were presented with *animations* about the operations of the binary heap that were similar to TRAKLA2 exercises (see Figure 2). In addition, student pairs in both conditions were given an exercise sheet that asked questions on binary heap that were supposed to be answered during the learning process. In this way, we tried to motivate the learning and make sure that the possible differences are due to controlled variable (level of engagement), and not because pairs in one condition performed cognitively more demanding activities or used more time on the tasks (Grissom et al., 2003; Hundhausen et al., 2002).

Method 2: Observational Study

The students' activities during the controlled experimental study were also recorded utilizing a screen capturing software. The recording accompanied by an audio track contained on-screen activity, i.e., mouse movements, keyboard typings, scrolling of the tutorial page back and forth in the browser window, as well as the conversation between the pair members.

The observed pairs were aware of being observed and we asked a permission to monitor them in advance. In this overt research method, we observed the students in their activities without intervention, i.e., by watching the recordings afterwards (Gall et al., 2006).

A detailed record of the events that occurred during the period of monitoring the students was produced. These events were categorized into the following four engagement levels according to the extended engagement taxonomy: *no viewing* (e.g., reading phase), *viewing* (e.g., watching figures), *controlled viewing* (e.g., watching of animations or model solution step-by-step with user controls) and *changing* (i.e., solving an algorithm simulation exercise). We separated passive *viewing* and more active *controlled viewing* from each other. In passive *viewing*, there was a still picture on the screen that we assumed the pair was watching. However, some of this time was spent to solve the given exercises on paper, as well. In *controlled viewing*, however, we knew that students were more actively involved with the animation as we required that they needed to control the animation by pressing VCR-like buttons to execute the animation backwards or forwards, and there were no pauses longer than 20 seconds between each action. The total time-on-task was measured from each four EET levels. Obviously, the students in *controlled viewing* condition (control group) did not spend time on *changing* mode. However, not all students in *changing* condition (treatment group) did either. Based on this analysis, we classified the students to groups based on their behavior.

Participants

Students were mainly first year students, however, some students from other years were also on the course. Students were randomized to the computer lab sessions and sessions were randomly assigned to each condition with the limitation that parallel sessions belonged to different conditions. The total number of participating students was 92. However, not all of them allowed to monitor their performance, nor were they willing to do pair work. In addition, in some of the workstations, the Java applet was not working properly. Moreover, we excluded foreign students from the study as they did not get the same treatment as the others due to the fact that their study materials were in a different language (i.e. English, while the original materials were in Finnish) and did not include animations nor algorithm simulation exercises, but they solved them by paper and pencil. Thus, the total number of analysis units (students) was 75 (n = 75) divided into 7 small groups (3 control groups having *viewing* condition and 4 treatment groups having *changing* condition). The original number of lab sessions was 8, but the last one (that would have been control group) was the excluded English speaking group.

All students had been previously using TRAKLA2 during the course to complete three assignment rounds related to basic data structures (e.g., lists and stacks), algorithm analysis, sorting algorithms (i.e., insertion sort, quicksort, and mergesort), and binary tree traversing. Thus, all students should have been able to use TRAKLA2, understand its visualization, and know all its features that were needed to complete the assignments.

Materials

Pre-test consisted of the following questions. In the first question, the student were asked to define concepts *array*, *binary tree*, and *priority queue*. We assumed that the students are able to answer the first two as those concepts were already introduced in the course. The last concept and the rest of the questions were such that we assumed the participants do not have prior knowledge to answer them. However, we wanted to test whether they have some prior knowledge, e.g., due to taking the course already in the previous year (without passing it). The second question was, if a given array is a heap and the third, whether an ordered array is a heap or not. In addition, we asked the students to describe where the smallest value in a minimum binary heap (question 5) and maximum binary heap is located (question 6), respectively. Finally, we asked them to write down a given binary heap's heap property (question 7). The third question asked the students to draw the binary tree representation of the minimum binary heap, which was given in an array presentation, in the previous question.

The post-test consisted of the following questions. The pre-test and post-test included two questions which were exactly the same. The first question in the pre-test was omitted from the post-test. However, the questions 2, 3, 4, 5, 6 and 7 were the same in both (but the numbering started from 1 in the post-test). In addition, participants needed to do similar exercises that they did in the lab session. One of these was insertion of new items into an initially empty maximum binary heap (question 7 in the post-test). The question 8 asked participants to remove two smallest items

from a minimum binary heap. Finally, we gave a pseudo-code example of a recursive `MAX-HEAPIFY` procedure and asked several questions, such as for which algorithm one can apply this procedure (question 9). This was a multiple choice question with four alternatives of which the last three were applicable: `Heap-Insert`, `Heap-Exctract-Max`, (linear-time) `BuildHeap`, and `HeapSort`. In addition, we asked them to describe and give an example execution (line-by-line) of what this procedure does and how (question 10). Question 11 requested the participants to provide an example which shows the recursive nature of the algorithm. The code example did not have a complete implementations for how to inquire the left and right child of a node in a complete binary tree implemented as an array. The task was to write this code (e.g., `LEFT(i) = 2i` and `RIGHT(i) = 2i+1`) (questions 12). Finally, they needed to analyze the worst case time complexity of `MAX-HEAPIFY` (question 13).

Procedure

Study was performed halfway through the course at the computer lab sessions that lasted for 2 hours. There were a total of 4 + 3 sessions, and they were run on two days in two following weeks. On each day, there were two times two sessions with different conditions running simultaneously. On the second day, there were also 4 sessions, but only 3 of them were included in this study as the last one was the excluded session given in English.

In the beginning of the session, students took the individual pre-test, in which they needed to answer questions related to binary heaps in 15 minutes. After this, they freely formed pairs with their peers and gave their consent to participate in the experiment and to be monitored during the experiment. If there was an odd number of students, one group consisted of 3 students. Each pair was allocated to a single computer.

After the pre-test, students had 45 minutes to go through the learning materials of their condition and complete paper-and-pencil exercises together. The collaboration was monitored by recording their talking and capturing their activities on the computer screens. After the 45 minutes the paper-and-pencil exercises were collected and the session ended with an individual post-test. The students were given 30 minutes to answer the questions in the post-test.

Each question in the pre- and post-tests was analyzed in a scale from 0 and 4. Zero points meant less than 25 percent of the answer was correct in the answer, and each point meant a 25 percent increase in the correctness of the answer.

Results

Randomized Treatment and Control Groups

In this section, we report the results as they were obtained by using the randomized treatment groups (42 students) and control groups (33 students) ($n = 75$).

Previous Knowledge and Motivation

All the information related to the previous knowledge of the students could be determined only through post-hoc analysis, and thus, we could not make sure before-hand that the randomization did not introduce any bias to the experimental settings. Table 1 represents the students' previous knowledge in Computer Science and Mathematics for both groups. The first column shows the pre-test scores for the topics studied in the experiment. The column "Prog. Course Results" shows the students' average grades from a previous programming course. The average number of CS and Math credits units (each credit unit equals to about 30 hours of work) obtained are shown in the next columns, respectively. The difference between groups in the previous programming course grades is approaching statistical significance ($t(73) = -1.94$, $p = 0.056$). Other differences are statistically insignificant.

Table 1: Previous knowledge of the students on Heap data structure, and in CS and Math

	Pre-test	**Prog. Course Grade**	**CS**	**Math**
Control (33)	9.27 (6.87)	2.61 (1.77)	10.72 (16.77)	9.13 (9.33)
Treatment (42)	8.57 (5.04)	3.36 (1.57)	10.44 (14.80)	8.34 (6.87)

Table 2 shows the results from a motivational questionnaire filled in by the students. The questions were answered in a 7-degree Likert-scale and they were as follows:
Q1. How useful do you regard this course for your working career?
Q2. Do you expect that the on-line learning will help your learning of the course content?
Q3. How well do online exercises fit into this course?
Q4. How useful have the on-line learning tools and materials been in your previous courses?

Table 2: Motivation of students based on a questionnaire. *Note*. Questions Q1 to Q4 are discussed in the text

	Q1	**Q2**	**Q3**	**Q4**
Control	4.84 (1.25)	4.78 (1.18)	5.38 (1.01)	4.94 (1.39)
Treatment	5.12 (1.33)	5.24 (1.14)	5.88 (1.05)	5.59 (1.30)

There were no statistically significant differences between the groups in any of the questions in the motivational questionnaire.

Post-test results

In the post-test, we used the same questions as in the pre-test and in addition to this seven more demanding questions. In the questions that were the same as in the pre-test, control and treatment group received on average 16.88 points (*st.dev.* 4.34) and 17.38 points (*st.dev.* 4.32), respectively. When comparing the pre- and post-test scores on the same questions within the group, statistically significant differences were found in both groups' total scores using pairwise t-test (Control: $t(33) = -13.48$, $p < .001$, Treatment: $t(42) = -25.71$, $p < .001$) (see the Table 1 for average pre-test scores and standard deviations). This means that both groups had learned the subject, which seems obvious when they spent 45 minutes to learn the topic.

When the points from all the questions were summed together the control group received on average a total of 30.79 points (*st.dev.* 6.99) and the treatment group 31.55 (*st.dev.* 6.29) points out of 52 points. There were no statistically significant differences found between the post-test scores.

We further calculated pair averages by taking the average of individual post-test scores of the pair. We treat this value as the learning outcome of a pair. The pair averages for control and treatment groups were 30.68 points (*st.dev.* 4.74) and 31.63 points (*st.dev.* 4.44), respectively. There were no statistically significant differences between the final scores or in any individual question scores.

Observational Study

In this section, we report the results as obtained by using a video analysis to match the students activities with the definition of treatment and control group. Based on the analysis, we regrouped students into different groups based on their behavior during the observation. We identified three groups based on their assignment to control and treatment groups and their behavior. Firstly, the students in the control group seemed to behave homogeneously and they watched the animations as expected. We will refer to this group with the name *Viewing C* (C as in Control). Secondly, we identified a group of students in the treatment condition, who behaved exactly the same as the control group by only watching the animations and not even once trying to do any algorithm simulation exercises. We will refer to this group with the name *Viewing T* (T as in Treatment). We will refer to all students who only viewed the animations (i.e. students in groups *Viewing C* and *Viewing T*) with the name *Viewing A* (A as in All. Thirdly, we found the students who behaved as we expected in the treatment group. These students solved algorithm simulation exercises at least one time but most often three to six times. We will refer to this group with the name *Changing T*. The division of the groups is illustrated in Figure 3.

Based on the video analysis, we classified 33 students to the *Viewing C*, 17 student to the *Viewing T*, and 21 students to the *Changing T* (n=71). We needed to exclude four students from the analysis in this section due to technical problems when matching the students to correct videos. Two of the students would have belonged to the *Viewing T* and two to the *Changing T* groups.

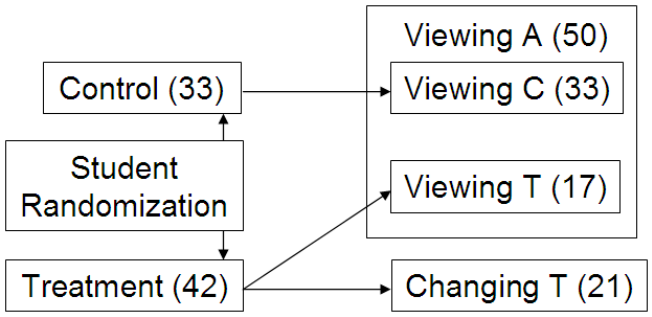

Figure 3: The division of the groups

In this section, we present two comparisons. Firstly, we analyze the data between three groups, namely *Viewing C*, *Viewing T* and *Changing T* because based on the original randomization and the video analysis these groups are distinct. However, when only the video analysis and groups' behavior is taken into consideration, we have only two groups, namely *Viewing A* and *Changing T*. Therefore, in order to provide a complete account of the results, we provide the analysis of both of these groupings. The validity, justifications and methodological implications of these groupings are further discussed in section **Error! Reference source not found.**.

Previous Knowledge and Motivation

The format of Table 3 is similar to the Table 1. None of the differences were statistically significant neither *Viewing C* vs. *Viewing T* vs. *Viewing T* nor *Viewing A* vs. *Changing T*. This was different compared to the original experimental design where there was a significant difference in favor of the treatment group in previous programming course grades.

Table 3: Previous knowledge of the students on Heap data structure, and in CS and Math

	Pre-test	Prog. Course Grade	CS	Math
Viewing C	9.27 (6.87)	2.61 (1.77)	10.72 (16.77)	9.13 (9.33)
Viewing T	8.06 (4.49)	3.47 (1.46)	12.56 (21.04)	7.69 (6.63)
Viewing A	8.86 (6.14)	2.90 (1.71)	11.33 (18.10)	8.64 (8.46)
Changing T	9.29 (5.72)	3.14 (1.80)	10.43 (9.35)	9.67 (7.21)

Table 4 shows the results from the same motivational questionnaire that was also reported in the Table 2 for the experimental groups (See Section 0 for the description of the questions). None of the differences were statistically significant.

Table 4: Motivation of students based on a questionnaire. *Note*. Questions from Q1 to Q4 are discussed in Section 0

	Q1	Q2	Q3	Q4
Viewing C	4.84 (1.25)	4.78 (1.18)	5.38 (1.01)	4.94 (1.39)
Viewing T	5.00 (1.51)	5.25 (0.93)	5.81 (1.05)	5.44 (1.26)
Viewing A	4.90 (1.32)	4.94 (1.12)	5.52 (1.03)	5.10 (1.36)
Changing T	5.19 (1.33)	5.19 (1.36)	5.86 (1.11)	5.67 (1.43)

Time Allocation between Engagement levels

Table 5 presents the distribution of the average times spent on each EET level. This was measured by watching the videos and marking times when the EET level changed from one to another, and then summing up the times on each EET level.

Table 5: The distribution of time (45 minutes) between EET levels

	No viewing	Viewing	Controlled viewing	Changing
Viewing C	47.45 % (15.28)	38.26 % (12.24)	14.29 % (6.23)	0.00 % (0.00)
Viewing T	49.45 % (17.09)	37.82 % (15.01)	12.73 % (5.47)	0.00 % (0.00)
Viewing A	48.13 % (15.78)	38.11 % (13.10)	13.76 % (5.97)	0.00 % (0.00)
Changing T	43.22 % (19.20)	38.30 % (15.84)	5.87 % (6.03)	12.61 % (1.98)

Table 6 shows how many times students used materials on each EET level. For example, students in the control group used user-controlled visualizations (*controlled viewing*) 5 times on average, whereas students in the treatment group used them 2 or 3 times on average.

Table 6: The number of times each EET level was used

	No viewing	Viewing	Controlled viewing	Changing
Viewing C	6.76 (2.11)	7.82 (3.61)	5.15 (2.71)	0.00 (0.00)
Viewing T	7.18 (2.19)	7.53 (3.04)	5.29 (2.91)	0.00 (0.00)
Viewing A	6.90 (2.12)	7.72 (3.40)	5.20 (2.75)	0.00 (0.00)
Changing T	6.24 (1.73)	6.67 (3.20)	2.48 (2.56)	4.10 (1.61)

Post-test results

The results of the post-test are presented in Table 7. When comparing the pre- and post-test scores within the group, statistically significant differences were found in both groups' total scores between pre- and post-tests when only same questions were compared with pairwise t-test (*Viewing C*: $t(32) = -13.15$, $p < .001$\$, *Viewing T*: $t(16) = -13.96$, $p < .001$, *Viewing A*: $t(49) = -18.09$, $p < .001$, and *Changing T*: $t(20) = -19.35$, $p < .001$) (see the Table 3 for average pre-test scores and the *subtotal* in the Table 7 for the comparable average post-test scores and standard deviations).

Table 7: Post-test results. *Note*. Post-test questions were discussed in Section 0 and compostion of the groups in Figure 3

	Viewing C	Viewing T	Viewing A	Changing T
Question 1	2.64 (1.58)	2.12 (1.65)	2.46 (1.61)	2.33 (1.80)
Question 2	1.76 (1.23)	1.82 (1.29)	1.78 (1.23)	2.19 (1.29)
Question 3	3.64 (1.08)	4.00 (0.00)	3.76 (0.89)	4.00 (0.00)
Question 4	2.39 (1.23)	2.18 (1.33)	2.32 (1.42)	2.33 (1.59)
Question 5	2.61 (1.43)	2.65 (1.58)	2.62 (1.47)	3.38 (0.92)
Question 6	3.85 (0.71)	3.76 (0.97)	3.82 (0.80)	4.00 (0.00)
Subtotal	16.88 (4.34)	16.53 (4.90)	16.76 (4.49)	18.24 (3.56)
Question 7	3.97 (0.17)	3.94 (0.24)	3.96 (0.20)	3.43 (1.29)
Question 8	3.33 (1.19)	3.65 (1.00)	3.44 (1.13)	3.76 (0.89)
Question 9	2.48 (0.87)	2.12 (0.78)	2.36 (0.85)	2.67 (0.91)
Question 10	2.09 (1.44)	2.41 (0.94)	2.20 (1.29)	2.62 (1.40)
Question 11	0.45 (1.25)	0.71 (1.45)	0.54 (1.31)	1.10 (1.70)
Question 12	1.30 (1.85)	0.18 (0.73)	0.92 (1.64)	1.24 (1.84)
Question 13	0.27 (0.45)	0.29 (0.99)	0.28 (0.67)	0.29 (0.46)
Total	30.79 (6.99)	29.82 (5.71)	30.46 (6.54)	33.33 (6.71)
Pair Average	30.68 (4.74)	29.88 (4.37)	30.42 (4.55)	33.45 (4.34)

Based on ANOVA, there were no statistically significant differences between *Viewing C*, *Viewing T* and *Changing T* groups in the post-test scores. When comparing the total values from the post-tests between *Viewing A* and *Changing T*, statistically significant differences were found in the total and pair average of the post-test scores by using one-tailed t-test ($t(69) = -1.73$, $p < 0.05$) and ($t(31) = -1.97$, $p < 0.05$), respectively.

Discussion

Interpretation of the Results

We presented an empirical study which analyzed whether the EET framework can be used to predict performance differences when algorithm visualizations are used in collaboration. Two randomized groups of students were involved in this study reading and answering questions related to a hypermedia tutorial presented on a web page. The control group used the algorithm visualizations on *controlled viewing* level, on which they had the opportunity to watch algorithm animations embedded in the tutorial. The treatment group interacted with the tutorial on *changing* level, on which they had the option to solve small algorithm simulation exercises and get feedback on their performance. In both groups, the students formed pairs and learned collaboratively about the binary heaps for 45 minutes during the 2-hour closed lab session. The analysis of the video material has showed that students were collaborating and discussing the subject matter during the learning process, therefore we are confident to say that students were truly learning collaboratively in both groups (Myller et al., in press). The null hypothesis of the experiment was that there would be no significant statistical difference between the learning outcomes of the control and treatment group after the session.

Pre- and post-tests were used to analyze the performance. Each student answered these tests individually. There were no significant differences between groups if we analyzed only the pre-test scores. However, post-hoc analysis of some background variables revealed that there was almost a significant bias between the groups. The grades from the previous programming course were better in the treatment group than in the control group. Furthermore, based on the post-test results we could not reject the null hypothesis. This all was (at first) a counter-intuitive result, because a) it was against the theory that we were testing, b) it was against our previous findings and c) even the bias between the groups was in favor of the treatment group.

Fortunately, during the experimental study, we monitored the student pairs in a parallel observational study. After examining the video recordings, we realized that not all of the students in the treatment group were using the tutorial as expected. Some of the pairs did not solve the exercises, but only watched the model solutions instead. Thus, they were interacting with the tutorial only on *controlled viewing* level, not in *changing* level as expected. Based on this new evidence, we re-grouped the students. We regarded those students in the treatment group, not behaving on the changing level, belonging to a controlled viewing level. Interestingly, the aforementioned bias in previous programming course grades disappeared, and we found significant differences between the learning outcomes of the groups. Although there were no differences when only three groups were compared, the group working on changing level outperformed all student groups working on controlled viewing level in the total score of post-test. This was true both in the individual performance and the average performance of pairs. Thus, based on this study, we can reject the null hypothesis and confirm our previous findings that the level of engagement on which the students interact with the visualization tool has an influence on the learning. On changing level, they learned better than on controlled viewing level.

Stasko et al. (1993) hypothesize that "algorithm animations will not benefit novice students just learning a new topic as much as the animations will benefit more advanced students", and moreover, that "the novice students would benefit more by actually constructing an algorithm animation rather than viewing a predefined one." We can confirm these hypotheses. However, in this first hypothesis, we need to be careful in the definition of a "novice". In our experiment, all students were exposed to TRAKLA2 before they attended the experiment. They solved similar exercises, but on different topics, a couple of weeks before the experiment took place. Thus, they were not "novices" when it comes to the "graphical notation" used in the experiment. Still, they were novices when it comes to the topic (i.e. they had not studied binary heaps earlier). Therefore, the conclusion is that the first hypothesis holds only if "novice" is defined to be a student who is not familiar with the used notation in the animations. One can still be a novice of the topic but understand the used notation, and benefit as much as more advanced students. Actually, it might even happen that the more advanced students cannot take the full advantage of this kind of learning material, and thus, perform worse, at least in relative scale (Myller et al., 2007). The confirmation of the second hypothesis is a direct outcome of our study in which the treatment group was "constructing an algorithm animation" in terms of changing the visualization, and they outperformed those students in the control group who just were "viewing a predefined" animation.

As discussed in the section on previous research, the learning time has not been a controlled variable in several previous studies, which have used the engagement level as the independent variable (Grissom et al., 2003; Naps et al., 2002; Hundhausen et al., 2002). Furthermore, it has been reported that students using visualizations on higher engagement levels have been motivated to spend more time on learning the topic. This has made it questionable if the time that students spend on learning the topic affects the learning results more than the engagement level, on which the visualization is used, and the engagement level affects only the amount of time students are willing to spend on learning the topic. In this study, we have shown that although we controlled the learning time and monitored students' activities, the learning results are significantly different between engagement levels. This means that the engagement level has a direct effect on the learning results.

Methodological Considerations

Based on the results, screen capturing and voice recording should be a standard procedure because we cannot always know for sure if the participants really do what we expect them to do. Our study shows that we could not have obtained full understanding of the phenomenon without monitoring the students: not all of them performed on the expected engagement level even though we instructed them to do so. As we can see from our study, the conclusion would have been that we could not find any evidence that the EET level has an impact on learning, which would have been a false negative result. Thus, monitoring should be a standard procedure especially in large scale studies in which the researcher(s) cannot make sure by other means that the conditions remain constant within a group.

However, when using an observational design in the study, we need to pay attention to possible confounds that might affect our results. Due to the fact that in the observational study, we could not control the placement of participants into conditions, but they selected it themselves, this could have caused differences in the final results and there still might be background variables that we have not analyzed or detected affecting the results. However, as stated earlier, we did a post-hoc analysis of several background variables and detected that actually the re-grouping made the groups more similar on one aspect while keeping the other aspects unchanged. Thus, we are fairly confident that the observed differences are due to the claimed causes.

Conclusion and Future Work

Our results confirm that EET framework can predict performance differences also in collaborative use of visualizations. The results substantiate that there is a difference in learning results between viewing and changing modes. The findings of the observational study also explain why the original experimental design failed to reject the null hypothesis. This was due to the fact that students in the treatment group did not perform the learning tasks that we assumed them to do. Thus, they might have outperformed the control group in the experimental design if they only had performed in the changing mode.

From our point of view, the results emphasize the importance of engagement with visualizations, and we should promote systems that support different modes of engagement. The mere viewing of the algorithm animations is not enough, not even when there is a partner with whom to share the understandings and misunderstanding during the viewing of the visualization. Thus, we should, especially, design systems that act on the higher levels of the engagement taxonomy. For example, visual algorithm simulation exercises acting on the changing level produce better results compared to the viewing level. Furthermore, we should encourage the use of the systems on higher engagement levels in classrooms in order to achieve active and more student-centered learning. We hope this paper encourages teachers on different disciplines to try out visualization tools that enable higher engagement between the tool and the students especially in collaborative learning as this seems to increase the learning outcomes.

The future research challenge is to determine the importance and role of collaboration in the EET, i.e., can we repeat this experiment also in the case of individual learning? In this experiment, collaboration was used to encourage discussion in pairs and to collect better evidence of the real behavior in terms of screen capturing. The collaboration, however, has an influence on the performance as well. Thus, one research direction would be toward individual learning, but in a context that can still be monitored in order to prevent inconclusive results due to the fact that the individuals did not behave on the expected EET level.

Acknowledgements

This work was supported by the Academy of Finland under grant numbers 111350 and 210947. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Academy of Finland.

References

Ben-Bassat Levy, R., Ben-Ari & M., & Uronen, P. A. (2003). The Jeliot 2000 program animation system. *Computers & Education, 40* (1), 15–21.

Byrne, M. D., Catrambone, R., & Stasko, J. T. (1996). Do algorithm animations aid learning? *Technical Report GIT-GVU-96-18*, Atlanta, GA: Graphics, Visualization, and Usability Center, Georgia Institute of Technology.

Collazos, C., Guerrero, L., Redondo, M., & Bravo, C. (2007). Visualizing Shared-Knowledge Awareness in Collaborative Learning Processes. *Lecture Notes in Computer Science, 4715*, 56-71.

Evans, C., & Gibbons, N. J. (2007). The interactivity effect in multimedia learning. *Computers & Education, 49* (4), 1147-1160.

Gall, M. D., Gall, J. P., & Borg, W. R. (2006). *Educational Research: An Introduction (8th Ed.)*, Upper Saddle River, NJ: Allyn & Bacon.

Grissom, S., McNally M., & Naps, T. L. (2003). Algorithm visualization in CS education: comparing levels of student engagement. *Proceedings of the First ACM Symposium on Software Visualization*, New York: ACM Press, 87–94.

Hübscher-Younger, T., & Narayanan, N. H. (2003). Constructive and collaborative learning of algorithms. *SIGCSE Bulletin, 35* (1), 6–10.

Hundhausen, C. D. (2002). Integrating Algorithm Visualization Technology into an Undergraduate Algorithms Course: Ethnographic Studies of a Social Constructivist Approach. *Computers & Education, 39* (3), 237–260.

Hundhausen, C. D. (2005). Using end-user visualization environments to mediate conversations: a 'Communicative Dimensions' framework. *Journal of Visual Languages and Computing, 16* (3), 153–185.

Hundhausen, C. D., & Brown, J. L. (2008). Designing, visualizing, and discussing algorithms within a CS 1 studio experience: An empirical study. *Computers & Education, 50* (1), 301–326.

Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. (2002). A Meta-Study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing, 13* (3), 259–290.

Kehoe, C., Stasko, J., & Taylor, A. (2001). Rethinking the evaluation of algorithm animations as learning aids: An observational study. *International Journal of Human-Computer Studies, 54* (2), 265–284.

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.

Korhonen, A., Malmi, L., Silvasti, P., Karavirta, V., Lönnberg, J., Nikander, J., Stålnacke, K., & Ihantola, P. (2004). Matrix - a framework for interactive software visualization. *Research Report TKO-B 154/04*, Helsinki: Department of Computer Science and Engineering, Helsinki University of Technology.

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

Laakso, M.-J., Salakoski, T., & Korhonen, A. (2005b). The feasibility of automatic assessment and feedback. *Proceedings of Cognition and Exploratory Learning in Digital Age*, Lisbon: IADIS Press, 113–122.

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2003). The impact of pair programming on student performance, perception and persistence. *Proceedings of the 25th International Conference on Software Engineering*, Los Alamitos, CA: IEEE Computer Society, 602–607.

Myller, N., Bednarik, R., Ben-Ari, M., & Sutinen, E. (In press). Extending the Engagement Taxonomy: Software Visualization and Collaborative Learning. *ACM Transactions on Computing Education*.

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

Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, New York: ACM Press, 359–362.

Naps, T. L., & Grissom, S. (2002). The effective use of quicksort visualizations in the classroom. *Journal of Computing Sciences in Colleges, 18* (1), 88–96.

Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., & Velázquez-Iturbide, J. Á. (2002). Exploring the Role of Visualization and Engagement in Computer Science Education. *Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education*, New York: ACM Press, 131–152.

Roschelle, J. (1996). Designing for cognitive communication: Epistemic fidelity or mediating collaborating inquiry. In Day, D. L., & Kovacs, D. K. (Eds.), *Computers, Communication & Mental Models*, London: Taylor & Francis, 13–25.

Scaife, M., & Rogers, Y. (1996). External cognition: how do graphical representations work? *International Journal of Human-Computer Studies, 45* (2), 185–213.

Stasko, J., Badre A., & Lewis, C. (1993). Do algorithm animations assist learning? An empirical study and analysis. *Proceedings of the SIGCHI conference on Human factors in computing systems*, New York: ACM Press, 61-66.

Suthers, D. D., & Hundhausen, C. D. (2003). An experimental study of the effects of representational guidance on collaborative learning processes. *Journal of the Learning Sciences, 12* (2), 183–219.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software, 17* (4), 19–25.