How Undergraduate Students' Learning Strategy and Culture Effect Algorithm Animation Use and Interpretation

Teresa Hubscher-Younger
Auburn University
Department of Computer Science and Software Engineering
107 Dunstan Hall
Auburn University, AL, 36849-5347
334-844-6342 (phone)
334-844-6329 (fax)
teresa@eng.auburn.edu

N. Hari Narayanan
Auburn University
Department of Computer Science and Software Engineering
107 Dunstan Hall
Auburn University, AL, 36849-5347
334-844-6312 (phone)
334-844-6329 (fax)
narayan@eng.auburn.edu

Abstract

Algorithm animation systems have not met the initial promise they seemed to hold for teaching algorithms to undergraduate computer-science students. A qualitative study of algorithm learning strategies of students and a usability study of a hypermedia algorithm animation system led to the conclusion that current algorithm animations may actually do more to hinder than help the undergraduate develop a solid understanding of the algorithm, because of undergraduates' learning strategies. Students relied on a single representation of an algorithm, misinterpreted the limitations and specifics of that representation, avoided using outside media sources and worked primarily within groups to understand algorithms. Thus, algorithm animation may have limited usefulness for this population. Therefore, we have students create, share and collectively interpret their own representations using learning technology systems to create more self-directed, engaged learners.

In Proceedings of the *First IEEE International Conference on Advanced Learning Technologies (ICALT 2001)*, IEEE Press, pp. 113-116.

How Undergraduate Students' Learning Strategy and Culture Effects Algorithm Animation Use and Interpretation

Teresa Hubscher-Younger
Auburn University
Department of Computer Science and Software Engineering
teresa@eng.auburn.edu

N. Hari Narayanan
Auburn University
Department of Computer Science and Software Engineering
narayan@eng.auburn.edu

Abstract

Algorithm animation systems have not met the initial promise they seemed to hold for teaching algorithms to undergraduate computer-science students. A qualitative study of algorithm learning strategies of students and a usability study of a hypermedia algorithm animation system led to the conclusion that current algorithm animations may actually do more to hinder than help the undergraduate develop a solid understanding of the algorithm, because of undergraduates' learning strategies. Students relied on a single representation of an algorithm, misinterpreted the limitations and specifics of that representation, avoided using outside media sources and worked primarily within groups to understand algorithms. Thus, algorithm animation may have limited usefulness for this population. Therefore, we have students create, share and collectively interpret their own representations using learning technology systems to create more self-directed, engaged learners.

1. Introduction

Abstract procedural concepts, such as algorithms, are notoriously difficult for undergraduate students to learn and difficult to teach. One cannot observe these algorithms directly, so many educators believe that using animation to visualize them would help students learn the concepts. Intuitively, it seems that algorithm animation should be easy to integrate into the learning practices of undergraduate students.

Currently, hundreds of algorithm animations and algorithm animation systems are available on the WWW. The goal of many of these systems is educational – to teach students new algorithms. The literature reports many educational benefits for algorithm animation, such as improving student motivation and developing student analytical skills [1].

However, an empirical study of learning an algorithm using animations by Stasko, Badre and Lewis [2] did not find a statistically significant difference in learning among students who used a textual description of an algorithm and those who used a textual description accompanied by an animation. Later studies, also have not found reliable and robust results showing learning benefits for undergraduate students viewing algorithm animation.

The evaluation of the systems have focused on either students' comments after using a system or whether students learn more from using the system than from more traditional methods of learning about the algorithm, such as listening to a lecture or reading a textbook [3]. What was not focused on was the type of problems undergraduate students typically have with learning algorithms and how do undergraduate students understand the representations given to them by the algorithm animations.

Algorithm animation systems are based on two implicit assumptions:

- Students work primarily with media, such as videos, textbooks and computer animations, in isolation (not in collaborative activity with peers) to understand algorithmic concepts
- The best representations to give the students to understand these concepts should be based on those representations and metaphors employed by experts in understanding an algorithm and communicating with other experts

A study of the current learning strategies students use for studying algorithms and a study of how students use an algorithm animation system show that these assumptions are flawed.

2. Current learning strategies

To discover how students learn algorithms a qualitative study was done with an introductory algorithm analysis class. The students were observed learning Quicksort, a sophisticated search algorithm, and were then interviewed about the strategies and tools they normally use to learn algorithms. First, the students were shown a videotaped lecture. Then they were given a set of questions to answer about that algorithm. The participants were told to bring any materials they normally use to study algorithms.

Six groups of two to three students were observed and videotaped, and 16 students have been interviewed, either in groups or individually.

The study found problems with both of the assumptions of algorithm animation systems. Most of the problems that students had in learning the algorithm had to do with their reliance on and misinterpretation of the representation of the algorithm presented in the video (a recursion tree diagram, a type of diagram commonly used by experts to explain recursive algorithms). Also, students avoided using media to learn the algorithm. They relied primarily on the lecture and their peers in understanding the algorithm, not outside media, such as textbooks.

2.1. Misinterpreted representation reliance

The main learning strategy employed by students was to imitate examples presented to them by the professor on the videotape. The students seemed to believe that if they worked examples, drawing diagrams similar to the ones they were given during the lecture, and thought hard about the diagrams, they would be able to answer the questions. They did this, even though not all the information they needed was contained in the representation of the algorithm.

Tree diagrams, such as the recursion tree diagram presented in the videotaped lecture, are often employed by experts to understand algorithmic problems. However, this diagram interfered with the students' understanding the algorithm. The problems with understanding the algorithm occurred mainly, because they did not understand the *limitations* of the representation. They believed it captured everything they needed to know about the algorithm, rather than just showing a partial view of the algorithm.

The students did not question the representation, understand that the representation is hiding complexity in the algorithm, or attempt to integrate new information with the representation. The partitioning strategy of the algorithm (how the algorithm divides the elements in the array for further sorting) was not well explained by the representation they were given. They did not realize the importance of this strategy and that they needed to learn more about it. They did not learn the partitioning strategy from the textbook, where it was explained in detail.

The test had several questions referring to this strategy and instead of seeking the answers from other sources they tried to derive a plausible explanation from a representation that did not contain any information that would be helpful with those questions. Most students did not seek out information from the textbook to supplement their understanding from the lecture, and if they did look in the textbook, most did not understand that the original array was changed by the partitioning strategy. The students often said in the interviews that they only use the textbook when studying right before the test to help them remember material in their notes.

When students were observed using the textbook, they became frustrated and were unable to integrate their knowledge from the lecture, their representation of the algorithm, and the description from the lecture. This might be due to the dense text in the textbook, which uses many mathematical symbols and comparisons to other algorithms.

Students were confused about concepts other than the partitioning strategy as well. There was confusion about the order of recursive calls and about the pivot picking strategy. These misconceptions might also have arisen from the students' reliance on *one* representation.

The students' attitude toward learning algorithms seemed to be that learning is something done in order to pass a test. They reported that they did not believe that the information learned in their classes would be applied outside of the classroom.

They seemed to also believe in an implicit contract between themselves and the teacher. In this fictional contract, the teacher tells them what exactly they need to learn, what tasks they need to do, such as homework, and what examples they need to be familiar with, and then the students are expected to know the material presented to them. They did not seem to engage in self-directed learning to learn about algorithms.

The students' reliance on one representation is dangerous, because they did not realize that the diagrams they were given were a *representation* of one main concept of the algorithm and that many different representations of the algorithm would also have explained it. They did not realize that the representation simplified the algorithm, emphasizing some aspects and deemphasizing (in fact hiding) other aspects.

2.2. Learning from your peers

Students were grouped together randomly for the study. However, the students in the study differed from each other considerably. There was a large group of Chinese graduate students in the class, and students differed also differed according to sex, race, work experience, previous schooling and age. The study grouped students together who had never met and who probably would not normally study together.

Despite this, only one group decided to have the members of the group work independently on the test questions. The students usually chose to work together and learn from each other, despite their differences. Most students reported liking working with each other in the session, and most reported regularly studying or working with friends to learn the algorithms and complete homework assignments.

"Groupthink" was a problem for many groups. Often the students would convince each other that explanations based on a faulty understanding of the algorithm were correct. They convinced each other that they did not need to use the textbook to answer the questions, and in one instance even convinced each other not to accept correct answers, when one student actually had decided to read the textbook during the session.

The belief of many algorithm animation system builders that students will use this system outside of class to learn and understand the algorithms is questionable. For the students in this study, unless a media source is well integrated into the lecture or classroom activities, it is not likely to be used.

3. Evaluation of hypermedia algorithm animation system use

A usability test of a hypermedia algorithm animation system HalVis [3] led to some unexpected conclusions. The usability test was expected to find problems with basic controls, such as navigation and animation controls. What was not expected was that the small study would find so many critical problems with how students interpret and understand the representations of the data structures and algorithm operations in the animation.

The study was done using eight students from another introductory algorithm analysis class. The students were observed using the system to learn the same algorithm, Quicksort. They were instructed to use a "think-aloud" protocol while using the system, so that the observer could understand what usability problems might be occurring when the student behaved in an unexpected manner.

The study found that all the students using the system misinterpreted at least one of the representations of the algorithm presented to them.

On the first screen, the students found a short, simple animation introducing the main concept of Quicksort: A group of cartoon men separate according to how they compare in height to the "pivot" man, and then the subgroups recursively do the same process.

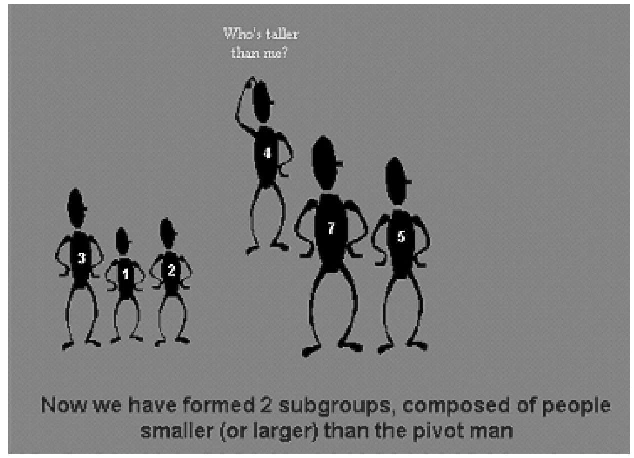

Figure 1: Image from animation introducing the main concept of Quicksort

Five of the eight participants had difficulty following the animation on this page. There seemed to be confusion about what was actually happening with the men on the screen and which dialog element belonged to which character.

The next screen with animation representing the algorithm had blocks with numbers representing different numbers of the array. The blocks moved around as they changed their position in the array. The blocks would turn different colors depending on their status.

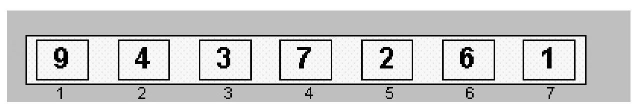

Figure 2: Starting array for animation of Quicksort

After watching the animation a few times, all but one of the participants reported that they felt they understood the animation. However, no participant could explain the color-coding scheme for the animation, when asked to do it. "I think the green numbers are less than the pivot, and the red are for numbers greater than the pivot. Wait it looks like they alternate color. I don't understand this at all," one participant said. The color-coding was critical to understanding the partitioning strategy of the algorithm

The third screen with animation representing the algorithm was meant to show how the algorithm works on a large data set. Thin bars represented data elements, and the height of a bar corresponded to the value of an element. Squares were drawn around the bars to represent which elements were included in a recursive call, and the color of the bars changed depending on their status.

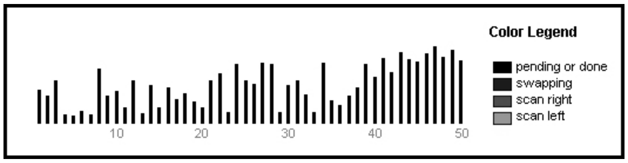

Figure 3: Image from animation of Quicksort sorting a large data set

Four participants did not understand the elements on the screen initially. One participant believed he was looking at a graph rather than data that would be sorted. He said, "I have no idea what this graph is supposed to be showing me. I wish it would tell me."

All of the participants, but one, reported that would like to be able to interrupt the animation and would like to slow it down, even though they had set it for “slowest speed”. Most participants blamed the speed for their inability to understand the animation, “I seriously have no idea what it’s doing. There’s no button to slow it down to give you a chance to figure it out.” It seemed like the students were trying to study each movement and step, rather than get an overall impression of how the algorithm works on large data sets, given different pivoting strategies and different types of data sets.

Presenting students with animations of the algorithm is supposed to help them be able to visualize the algorithm. However, interpreting the representation they were given to aid them can often be difficult work for an undergraduate.

4. Current Work

The learning strategies and culture of the undergraduate classroom observed in these qualitative studies imply that algorithm animations can be confusing, misinterpreted and unused by the undergraduate computer-science student.

Instead, we are proposing that students should engage in an active process of algorithm representation creation, sharing and collective interpretation. Students are more likely to accept representations as being incomplete and partial when created by their peers, rather than an authority figure, and thus may be better able to understand that there are different aspects of the algorithm that need to be understood. By sharing their representations, they will be better able to check their understanding with others. Collective interpretation will hopefully help correct misinterpretation problems and help an instructor see common misunderstandings.

The system we are building to help students do this, CAROUSEL (Collaborative Algorithm Representations Of Undergraduates for Self-Enhanced Learning) was used in a small pilot study with 12 students in a beginning data structures course. The system collected and displayed student-created representations and collected the ratings students gave for certain characteristics of these representations. Students in the pilot study chose to work with a wide variety of media, including text, graphics, sound and animation, based on their personal preferences.

Students were tested before they created representations and after to measure their knowledge of the algorithm. Initial results from the pilot study suggest that this activity does help learning. For two of the three algorithms that were used in the pilot study, there was a significant positive correlation between creating and sharing a representation and test scores ($r=.635$, $p=.07$; $r=.663$, $p=.05$), compared to students who did not engage in this extra activity.

5. Conclusions

As long as the technologies aimed at helping undergraduate students learn fail to address students' current learning strategies and culture, they will not be used or, worse, will confuse the student.

Evaluating current student learning strategies, observing students using the system and listening to their understanding of the content presented to them is invaluable to any developer of advanced learning technologies. To insure that a learning technology will actually be used, understood and helpful, one needs to understand the needs, culture and strategies of the learners.

6. References

[1] Gloor, P. *Elements of Hypermedia Design: Techniques for Navigation and Visualization in Cyberspace,* Birkhauser, Boston, 1997.

[2] J.T. Stasko, A. Badre, and C. Lewis, "Do algorithm animations assist learning? An empirical study and analysis", *Proceedings of the INTERCHI ’93 Conference on Human Factors in Computing Systems,* Amsterdam, Netherlands, 1993, pp. 61-66.

[3] Hansen, S. R. *A framework for animation-embedded hypermedia visualization of algorithms*, PhD Dissertation, Department of Computer Science & Engineering, Auburn University, Auburn, AL, 1999.