An evaluation of the effortless approach to build algorithm animations with WinHIPE

Jaime Urquiza-Fuentes and J. Ángel Velázquez-Iturbide
ViDo Research Group, Rey Juan Carlos University, Móstoles, Spain

{jaime.urquiza,angel.velazquez}@urjc.es

1 Introduction

Algorithm visualizations (AV) are used in computer science education since the early eighties [Baecker and Price, 1998]. There are various surveys on using visualization as an aid for computer science education [Grissom et al., 2003, Hundhausen et al., 2002, Naps et al., 2003a,b, Stasko and Lawrence, 1998]. In spite of their educational potential, they have not been incorporated into the mainstream of computer science education. This lack of use has two main reasons: from the instructors' point of view, animations are not usually easy to use, deploy and adapt to the course [Naps et al., 2003a, Ihantola et al., 2005]; from the students' point of view, more interaction, than just viewing animations, is needed to obtain learning improvements [Grissom et al., 2003, Hundhausen et al., 2002, Naps et al., 2003b].

Naps et al. [2003b] defined a taxonomy of engagement levels for the different ways of interaction between the students and the animations. The pedagogical effectiveness of these engagement levels in AV has been analyzed [Grissom et al., 2003, Hundhausen et al., 2002, Naps et al., 2003b]. The general conclusion is: the higher the engagement of students with AV technology, the better the learning outcomes. For instance Grissom et al. [2003] found learning improvements, at the understanding level of Bloom's taxonomy [Bloom et al., 1959], with AV extended with stop-and-think questions, the responding engagement level. We have found in the literature neither studies about AV technologies improving learning further than the understanding level of Bloom's taxonomy, nor studies about engagement levels in AV further than responding.

We have developed an effortless approach to build and maintain algorithm/program animations [Urquiza-Fuentes and Velázquez-Iturbide, 2005a]. Thus we have extended the WinHIPE IDE with visualization facilities to produce web-based algorithm/program animations. These animations consist of four components: the animation itself (a sequence of visualizations), the source code, the description of the algorithm implemented, and the description of the problem solved by the algorithm.

The main aim is to minimize the work needed to produce the animation. Animations are built from a set of static visualizations representing the execution stages of a program, which are automatically generated. Apart from the typical edition-compilation-execution process, the user will have to select which visualizations will form part of the animation, and type the problem and algorithm descriptions. We have designed an information visualization technique called R-Zoom [Urquiza-Fuentes and Velázquez-Iturbide, 2005b] that helps on the task of selecting the visualizations, and typing text is not a complex task. Thus, we have an animation generation process very similar to the edition-compilation-execution process of a program, where the additional tasks are not complex; this is why we call it an effortless approach.

Following the framework provided by the engagement levels, we have conducted a controlled evaluation to test if our effortless approach of building AV (construction engagement level) improves learning. In our evaluation, we have compared the viewing engagement level against the constructing engagement level. Learning improvement has been measured in terms of the comprehension and application levels of Bloom's taxonomy. We have completed this evaluation with measurements of efficiency and students' satisfaction.

The rest of the paper is structured as follows. Section two describes the evaluation:

participans, variables studied, method and produre. Then results of the evaluation are shown in section three. Finally, in section four, conclusions and future work are described.

2 Description of the evaluation

This evaluation attempts to find if there is any performance difference between viewing algorithm visualizations (viewing engagement level) and constructing algorithm visualizations (constructing engagement level). Improvements will be measured in terms of Bloom's taxonomy, with tasks related to the comprehension and application levels.

The context of this evaluation is an Algorithm Design and Analysis course at the Rey Juan Carlos University, where a group of students had to build algorithm animations using our effortless approach. In the evaluation, the tree breadth traversal algorithm was used.

2.1 Participants

Fifteen different subjects participated in the evaluation, thirteen were male and two female. Participation in the evaluation was voluntary. All of the subjects were students from the Algorithms course.

Participants were randomly divided in two groups: the control group and the experimental group (CG and EG respectively for the rest of the paper). Both groups were asked about their previous knowledge about the algorithm, only one student having previous knowledge. Therefore both groups belong to the same population and further results can be compared.

2.2 Variables

The independent variables of the evaluation were: pedagogical effectiveness and efficiency, and users' opinion about both approaches (viewing and building). The dependent variables were: the answers to a number of questions about the algorithm (mapped to the comprehension and application levels of Bloom's taxonomy), the time expended in studying the algorithm, the time used to complete a knowledge test about the algorithm, and the user's subjective opinion.

Learning improvements related to the comprehension level were measured with the following questions:

- What are the main ideas of this algorithm?
- Given the following tree, write the result of applying the algorithm to it.
- Given the following list, write the tree to which the algorithm was originally applied (note that the tree was balanced).
- Which is the existing relationship among the nodes of the tree, if the result is a strictly ascending ordered list?

Learning improvements related to the application level were measured with the following question: What modifications should be done on the algorithm to change the traverse direction to right-to-left?

Users' opinion was measured with a questionnaire where students were asked:

- if they thought that *building* (or *viewing*) algorithm animations had helped them in understanding the algorithm, and
- if they though that algorithm animations are easy to *build* (or *use*) with WinHIPE.

2.3 Method and procedure

The evaluation was divided into two sessions: a training session where the IDE was shown to the students, and the experimental session where knowledge about the algorithm was evaluated. Participation was ten and thirteen students respectively.

The training session was two hours long. The instructor demonstrated the tool, he generated two web-based animations with WinHIPE as an example, and students generated two more animations. The animations used were unrelated to the algorithm that would be used in the experimental session. None of the students appeared to have problems using the tool. At the end of this session a questionnaire about the tool was completed by the students thus, we got their first impression about the tool.

The experimental session also was two hours long, and two weeks after the training session. First, we explained to the students that we were carrying out the evaluation, and that their participation would be voluntary. Next, we randomly formed the CG (n=7) and EG (n=6), and we checked that all students in the EG had attended the previous training session. Students of both groups were asked about their previous knowledge about the algorithm. Then, we gave the students all the materials they were allowed to use to study the algorithm, which was a textual description of the algorithm for both groups, and:

- a number of web-based algorithm animations to be viewed, built with WinHIPE, for the CG, and
- the source code of the algorithm, to build web-based algorithm animations, for the EG.

Students of both groups were asked to study the algorithm using the materials, until they thought that they had enough knowledge about it. Then, they completed the knowledge test and another questionnaire to collect their subjective opinion about their viewing/building learning experience.

3 Results of the evaluation

Learning effectiveness was tested by means of two levels of Bloom's taxonomy: comprenhension and application. Both levels were graded in the range $[0.0, 1.0]$. Four questions were asked to test performance related to the *comprehension level*. The first question asked students to identify the main ideas underlying this algorithm; these ideas were: (1) operations with lists, (2) left-to-right direction in tree traversing, and (3) accumulation of recursive operations with subtrees. Students of both groups performed the same identifying ideas (1) and (2), while idea (3) was only identified by 14% of students (1/7) in the CG, but by 83% of students (5/6) in the EG ($p = 0.013$). We did not find significant differences in performance in the second, third and fourth questions. Thus, the average grades for the understanding level were 0.88 for the EG and 0.73 for the CG, a 16% of learning improvement.

We found significant differences ($p = 0.03$) in the answers to the question related to the *application level*. The CG obtained an average grade of 0.33, while the EG obtained an average grade of 0.77, a 60% of learning improvement.

We also measured the time expended with the materials and the time used to complete the knowledge test. With respect to the time expended by students using the materials to study the algorithm, students of the EG expended an average time of 49 minutes, while students in the CG expended an average time of 18 minutes. Difference of this time between groups was significant ($p < 0.05$). But no differences were found ($p = 0.194$) in the time used to complete the knowledge test between the EG and CG.

The students' first impression (after the training session) about WinHIPE and the building process was very good. None had previously used WinHIPE. All of them ($n = 10$) thought that the web-based animations were easy to build, and that building animations would help them in understanding the algorithms.

This opinion was maintained by students after the experimental session. Answers to the questionnaire about users' satisfaction showed that students in the EG agreed with both ideas: building algorithm animations helped them understanding the algorithm, and web-based animations were easy to build with WinHIPE. All of the students in the CG agreed with: web-based animations helped them in understanding the algorithm, and web-based animations were useful and easy to use. We also asked these students about what approach would be more helpful in learning algorithm concepts: viewing or building. 71% (5/7) thought that both approaches were equally helpful: two of them just said this, but three also said that both approaches should be used together.

4 Conclusions and future work

We have developed an effortless approach to build and maintain algorithm animations, and we have made a short term evaluation of it. This evaluation compares the learning improvements achieved with two engagement levels: viewing (viewers) and constructing (builders), where our approach is used. Learning improvements were measured in terms of two levels of Bloom's taxonomy: understanding and application.

Results show that, at the understanding level, builders obtained slightly better results than viewers, 16% of improvement; at the application level, builders improved learning the 60% more than viewers. Builders expended much more time than viewers, but they did not complain about it: they thought it was necessary and fruitful to use this amount of time.

According to the students' opinion, builders believed that building algorithm animations had helped them in understanding the concepts of the algorithm, and thought that animations were easy to build. Most of the viewers thought that viewing and building were equally helpful, but half of them thought that both engagement levels should be used together.

We realize that the generalization of these results is limited because of: the low number of students (15), the short period of time evaluated (two sessions of two hours), and the topic used (just the tree breadth traversal algorithm). But we think that this is a promising result because: we have empirical evidence of learning improvements at the understanding and application level of Bloom's taxonomy, using our effortless approach together with the building engagement level; and students felt comfortable with this approach, perceiving it as effective and helpful.

As future work, we plan to conduct a long term evaluation of this approach with functional program animations; also, effortlessness have to be evaluated from the instructor's point of view [Ihantola et al., 2005] so, we will evaluate the usability of the building/management process of these web-based algorithm/program animations and collections of them.

5 Acknowedgements

This work is supported by the research project TIN2004-07568 of the Spanish Ministry of Education and Science. Also authors want to thank: Carlos Lázaro-Carrascosa for his valuable help during the experimental session, and students of the Algorithm Design and Analysis course of the Rey Juan Carlos University for participating in the evaluation.

References

R. Baecker and B. Price. The early history of software visualization. In J.T. Stasko, J. Domingue, M.H. Brown, and B.A. Price, editors, *Software Visualization*, pages 29–34. MIT Press, Cambridge, Massachusetts, USA, 1998.

B. Bloom, E. Furst, W. Hill, and D.R. Krathwohl. *Taxonomy of Educational Objetives: Handbook I, The Cognitive Domain*. Addison-Wesley, 1959.

S. Grissom, M.F. McNally, and T. Naps. Algorithm visualization in CS education: comparing levels of student engagement. In *Proc. 2003 ACM Symposium on Software Visualization*, pages 87–94, New York, NY, USA, 2003. ACM Press.

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

P. Ihantola, V. Karavirta, A. Korhonen, and J. Nikander. Taxonomy of effortless creation of algorithm visualizations. In *ICER '05: Proc. 2005 International Workshop on Computing Education Research*, pages 123–133, New York, NY, USA, 2005. ACM Press.

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

T. Naps, G. Rößling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J.Á. Velázquez-Iturbide. ITiCSE 2002 working group report: Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, 35(2):131–152, 2003b.

J.T. Stasko and A. Lawrence. Empirically assessing algorithm animations as learning aids. In J.T. Stasko, J. Domingue, M.H. Brown, and B.A. Price, editors, *Software Visualization*, pages 419–438. MIT Press, Cambridge, Massachusetts, USA, 1998.

J. Urquiza-Fuentes and J.A. Velázquez-Iturbide. Effortless construction and management of program animations on the web. In Rynson W.H. Lau, Qing Li, Ronnie Cheung, and Wenyin Liu, editors, *Advances in Web-Based Learning - ICWL 2005, 4th International Conference*, volume 3583 of *LNCS*, pages 163–173, Berlin, Germany, 2005a. Springer Verlag.

J. Urquiza-Fuentes and J.A. Velázquez-Iturbide. R-zoom: A visualization technique for algorithm animation construction. In *Proc. IADIS International Conference Applied Computing 2005*, pages 145–152. IADIS Press, 2005b.