



ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

**Electronic Notes in
Theoretical Computer
Science**

Electronic Notes in Theoretical Computer Science 224 (2009) 169–178

www.elsevier.com/locate/entcs

Pedagogical Effectiveness of Engagement Levels - A Survey of Successful Experiences

Jaime Urquiza-Fuentes¹ and J. Ángel Velázquez-Iturbide²

*Department of Computer Science Languages and Systems - I
Universidad Rey Juan Carlos
Móstoles (Madrid), Spain*

Abstract

In this paper we survey experiments with program and algorithm visualizations (PAVs) where learning improvements have been detected. We analyze these experiments based on the student's level of engagement with the visualizations. There are some features present in most of these, successful, experiments. Therefore they should be taken into account as important factors affecting pedagogical effectiveness of PAVs, these features are: narrative and textual contents, feedback to students' answers and a student centered approach when designing PAV construction kits.

Keywords: Program visualization, algorithm visualization, engagement levels, educational evaluation, survey

1 Introduction

Studies about pedagogical effectiveness of PAVs have shown mixed results. The most significant result was reported by Hundhausen et al. [11], stating that the effort dedicated by students in visualization related tasks was more important than visual contents shown by PAVs. They also identified lack of research in some areas, e.g. using narrative and textual contents integrated with PAVs.

Following the idea of going beyond the passive viewing of PAVs, Naps et al. [20] developed a taxonomy that identified different ways of interacting with PAVs. They called the *engagement levels* taxonomy, and they suggested a hierarchical structure where more engagement should produce educational improvements.

After reviewing PAV literature, and research on some engagement levels [22,23], the authors feel that educational improvements could depend on other features. This survey studies possible effects of these features.

¹ Email: jaime.urquiza@urjc.es

² Email: angel.velazquez@urjc.es

The rest of the paper is organized as follows. First, in section 2, we describe the study we have carried out, the kind of papers included and their surveyed features. Then, in section 3 we detail the successful experiences, grouped by the engagement levels where the improvement were detected. In section 4 we analyze these experiences from two different point of views. Finally, in section 5 we draw our conclusions and future work.

2 The survey

Literature about PAVs with educational aim is wide, we have focused on successful experiences. These experiences must have detected educational improvements – knowledge acquisition, attitude towards the subject or materials, or programming performance– where PAVs have been used.

Having a look at published experiments, one suspects that just visualizations are not enough to obtain educational improvements. In fact, one of the most significant studies on PAV [11] concludes that the way that students use visualizations is more important than what visualizations show to students. Also, there are successful experiences based on providing high quality contents with the visualizations [7], advanced manipulation interfaces [5], or adding visualization sessions to regular classes [19].

Our aim is to deepen the effect that these additional features have on the educational improvements. The features that we have taken into account are:

Narrative contents and textual explanations They could help students to understand graphical depictions generated by a PAV system. In addition, when students build their own animations, adding narrative contents engage students in a reflection exercise that could produce learning outcomes.

Feedback on student's actions During animations, students can be asked to predict future steps of the algorithm. Feedback to their answers could reinforce right answers or correct wrong ones. As animations provide inherently feedback in the next step, we will take into account only explicit feedback, for both right and wrong answers.

Extra time using PAV Many tasks in typical learning environments can not be replaced with animation based tasks, therefore to use animations extra time is needed.

Advanced features Some systems provide with advanced contents showing different behaviors of the algorithm, advanced interfaces to manipulate visualizations, or advanced integration with an IDE.

Obviously, we have used the educational improvement reported by each experience and the engagement levels used. Educational improvements can be detected as knowledge acquisition, student's performance when programming their own solutions or student's attitude towards subjects or materials (usually knowledge acquisition is affected by attitude). We have detected two experimental designs, on the one hand there exist experiments studying improvements on one engagement level

or a mixture of two of them, on the other hand there exist comparative studies.

3 Successful Experiences

We have considered 24 experiences in this survey. In this section, we describe them grouped by the engagement level where the educational improvement has been detected. Table 1 summarizes these experiences.

3.1 Viewing

“Viewing” can be considered the core form of engagement, (...) a learner can view an animation passively, but can also exercise control over the direction and pace of the animation, use different windows (each presenting a different view), or use accompanying textual or aural explanations. (...) The remaining four categories all include viewing. [20]

The six experiences related to this level have detected educational improvements in terms of knowledge acquisition. The seventh chapter of Lawrence’s dissertation [17] detected improvements when using PAVs with textual labels. Crosby and Stelovsky [4] detected improvements when using multimedia materials made up of visualizations and narrative contents, comparing it with the *no viewing* level.

Kann et al. [13] made a comparative study among *no viewing*, *viewing*, *constructing* and, *viewing and constructing*. But they only detected significant improvements between *viewing* and *no viewing* students. It is the only viewing experience without textual or narrative contents.

Kehoe et al. [14] studied the use of PAV in a homework simulation environment, thus students used animations to complete the assignments without time limit.

Kumar’s experience [15] represents an auxiliary use of visualization. The main role of Kumar’s system is tutoring students providing them with automatic generated problems. His experience found that using visualizations within the feedback provided by the tutor improves knowledge acquisition.

Finally, Urquiza-Fuentes [22] investigates the effect of replacing part of exercises sessions with program visualizations sessions during a long term evaluation. The animations had additional textual explanations.

3.2 Responding

“Responding”. The key activity in this category is answering questions concerning the visualization presented by the system. (...) In the responding form of engagement, the learner uses the visualization as a resource for answering questions. [20]

The three studies of this level compare *responding* with *no viewing* level. The two first experiences detected improvements in knowledge acquisition and were supported by additional narrative contents. Although Byrne et al. [3] used a plain algorithm animation, the instructor provided the students with questions that had

to be answered during the animation. While Grissom et al. [6] used a system that integrated automatically the questions within the animation.

Finally, Laakso et al. [16] went beyond simple questions, engaging the students in simulation tasks. Here, the students manipulate a data structure simulating the behavior of a given algorithm, receiving explicit feedback about their simulations. But they also used the *viewing* level, as the students were allowed to see animations related to the algorithm that they had to simulate.

3.3 Changing

“Changing”, entails modifying the visualization. The most common example of such modification is allowing the learner to change the input of the algorithm under study in order to explore the algorithms behavior in different cases. [20]

The two first experiences mixed *responding* plus *changing* levels, and compared them with *viewing* and *no viewing* levels. They can be found in the same publication Hansen et al. [7] –studies I, II, IV and V–. Instead of using just isolated animations with additions, they produce high quality materials providing the students with three different animations –conceptual/abstract, detailed and populated– of the same algorithm, asking questions to the students and providing them with explicit feedback.

Lawrence studied the effect of changing input data to animations against *no viewing* and *viewing* levels. In the comparative study with the *no viewing* level [18] she found improvements in knowledge acquisition; the animations had narrative contents and students who worked with them had an additional lab session. She also compared this level with the *viewing* one [17], obtaining again improvements in knowledge acquisition without additional features.

Ben-Bassat et al. [2] studied the use of a visualization tool for teaching novices java. They found that only medium students improved their knowledge. Moskal et al. [19] focused on novice students “at risk” of not succeeding in their first programming course. They detected improvements in knowledge acquisition with an extra subject where students worked with an advanced tool to learn OO programming basics.

Ahoniemi and Lahtinen [1] compared this level with the *no viewing* level. They used animations with additional narrative contents. This experience used homework assignments, therefore working time was not limited.

The last changing experience [5] found improvements in programming performance. The instructors provided students with an advanced tool integrated in a Java IDE, while the students in the *no viewing* group used the same environment without visualization features. The students completed programming and debugging tasks with the environment.

3.4 Constructing

“Constructing”. In this form of engagement, learners construct their own visualizations of the algorithms under study. Hundhausen and Douglas [27] have

identified two main ways in which learners may construct visualizations: *direct generation* and *hand construction*. (...) It is important to note that the *Constructing* form of engagement does not necessarily entail coding the algorithm. [20]

Stasko [21] designed assignments where students had to construct their own animations. This also included some *changing* activities. He detected that students dedicated more time to study those algorithms for which they had constructed animations.

Urquiza-Fuentes and Velázquez-Iturbide [23] made a short term comparative study with the *viewing* level. Students within the *constructing* group generated animations with textual explanations using an effortless approach, while the others just viewed the same kind of animations, thought generated by the instructors. They detected improvements in students' attitude, *constructing* students remained studying the algorithm more time, and their knowledge acquisition was improved.

Finally, Urquiza-Fuentes [22] studied the effect of the same construction approach in a long term evaluation. He compared the *constructing* level with *viewing* and *no viewing* levels. He detected improvements in attitude on both comparisons; he also detected improvements in knowledge acquisition when comparing with the *no viewing* level.

3.5 Presenting

“Presenting”, entails presenting a visualization to an audience for feedback and discussion. [20]

The three experiences studying presenting level include construction tasks, therefore all have additional narrative contents. Two of them have focused just on this mixture of tasks [9,10], while the other compared it with the *viewing* level.

First, Hundhausen [9] compared constructing and presenting tasks using two different tools: a wellknown algorithm visualization tool, and utilities selected by the students – ranging from slides to crafts–. This observational study detected improvements in attitude of those students who used their own utilities. Using these results, a tool for algorithm animations construction was designed and compared again with construction utilities selected by the students [10]. In this experience, improvements in programming performance were detected on the students who worked with the designed tool.

Finally, Hübscher-Younger and Narayanan [8] compared *presenting* and *constructing* levels with the *viewing* level. They encouraged students –voluntary task– to generate animations and asked them to evaluate –compulsory task– those generated by the rest of the students. The construction utilities were chosen by the students. They detected improvements in knowledge acquisition of the students who constructed the animations.

Table 1 Summary of successful experiences grouped by engagement levels

Experience	Educational improvement	Engagement levels	Narrative contents	Explicit feedback	Extra time	Advanced features
Lawrence [17] Ch.7	Knowledge acq.	(V)	★			
Crosby and Stelovsky [4]	Knowledge acq.	(V) ⇒ NV	★			
Kann et al. [13]	Knowledge acq.	(V) ⇒ NV			★	
Kehoe et al. [14]	Knowledge acq.	(V) ⇒ NV	★			
Kumar [15] (Feedback of tutoring system)	Knowledge acq.	(V) ⇒ NV	★			
Urquiza-Fuentes [22]	Knowledge acq.	(V) ⇒ NV	★			
Byrne et al. [3]	Knowledge acq.	(R) ⇒ NV	★			
Grissom et al. [6]	Knowledge acq.	(R) ⇒ NV	★	★		
Laakso et al. [16]	Attitude	(R,V) ⇒ NV				
Hansen et al. [7] Studies I-II-IV	Knowledge acq.	(CH,R) ⇒ NV	★	★		Contents
Hansen et al. [7] Study V	Knowledge acq.	(CH,R) ⇒ V	★	★		Contents
Lawrence [17] Ch.6	Knowledge acq.	(CH) ⇒ V	★		★	
Lawrence et al. [18]	Knowledge acq.	(CH) ⇒ NV	★			
Ben-Bassat et al. [2]	Knowledge acq.	(CH)			★	IDE interface
Moskal et al. [19]	Knowledge acq.	(CH) ⇒ NV			★	Vis. interface
Ahoniemi and Lahtinen [1]	Knowledge acq.	(CH) ⇒ NV	★		★	
Cross et al. [5] & Jain et al. [12]	Prog. perform.	(CH) ⇒ NV			★	
Stasko [21]	Attitude	(C,CH)				
Urquiza-Fuentes and Velázquez-Iturbide [23]	Att./Know. acq.	(C) ⇒ V	★		★	
Urquiza-Fuentes [22]	Att.&Know. acq.	(C) ⇒ NV	★			
Urquiza-Fuentes [22]	Attitude	(C) ⇒ V	★			
Hundhausen [9]	Attitude	(P,C)	★			
Hübischer-Younger and Narayanan [8]	Knowledge acq.	(P,C) ⇒ V	★			
Hundhausen and Brown [10]	Prog. perform.	(P,C)	★			

4 Discussion

4.1 A global view

Clearly, learning can be enhanced with PAV. The 75% (18/24) of the experiences have detected improvements in terms of knowledge acquisition, together with more than 20% (5/24) detecting improvements in attitude towards the materials used or the subjects affected by the study. Finally, programming skills can also be improved, as they have been detected in more than 8% (2/24) of experiences.

Looking at the successful engagement levels investigated, there are two ends. *Changing* is the most investigated level with the 37.5%(9/24) of the experiences, while *presenting* is the opposite with 12.5%(3/24). *Responding* is present in the 20.8%(5/24) of experiences, and both *viewing* and *constructing* are present in the 27.2%(7/24) of experiences.

Not all experiences compare two different levels, 20.8%(5/24) of them explore possible improvements within a concrete level. When looking at the comparative experiences, the 73.7%(14/19) have studied the PAV effectiveness against no use of it, the rest –26.3%(5/19)– did it against the *viewing* engagement level.

The use of narrative and textual contents is present in the 75%(18/24) of the experiences. This means that they are an important factor to take into account when designing learning experiences with PAV. While explicit feedback, extra working time or advanced features –high quality contents, advanced interfaces– are present on more than 20% of the experiences.

4.2 Recommendations for designing visualization based learning experiences

As this is not a meta study like [11], we can not give formal and scientific evidence of correlations among different engagement levels and educational improvements. But all these experiences give empirical evidence on successful uses of different engagement levels, thus we can extract a number of recommendations for each engagement level.

Just viewing animations can improve knowledge acquisition, but animations should have additional text or narrative contents.

When students **answer questions during the animation**, again they should be provided with additional narrative or textual contents. But explicit feedback is also important, although it is not used in two of the experiences, the questions used in these experiences were predictive ones, thus the correct answer is given in the next steps of the animation.

Allowing the students to **change input data** is a more active task. Here, narratives and textual contents seem to be less important 62.5%(5/8). The reason could be that researchers were more interested in cognitive work performed by students when choosing input data, rather than explaining students what happens. As this is an explorative task, a strict time limit should be avoided. But also some advanced features as high quality contents –different execution conditions [7]–, the integration with the IDE [19], or the interface used to manipulate animations [5,12],

could produce learning outcomes.

When students **construct their own animations**, the construction interface is very important. Thus, providing the students with carefully designed interfaces, or allowing them to choose their own construction kits, have been shown to be effective³. Encouraging students to produce their own textual or narrative contents is also positive. Here, most improvements have been detected in attitude towards materials and subjects.

Finally, when students are asked to **present animations**, they also should construct them. Therefore, the construction interface is important again.

4.3 Suggestions for moving among engagement levels

Looking at the experiences, we can analyze what engagement levels have been overcome by others and how. Most of the experiences report on improvements when comparing with the *no viewing* and *viewing* engagement levels.

Coming from the *no viewing* engagement level

The *no viewing* level means that no PAVs are being used. Thus, a simple change is to move to the *viewing* level, where knowledge acquisition is improved. It can be a simple movement because there exist a number of PAV collections, but if one wants to generate her own PAVs, the narrative and textual contents should be taken into account.

Moving to the *responding* level is also possible because, again, there are existing PAV collections. This movement can improve attitude and knowledge acquisition. When designing your own *responding* experiences the use of narrative contents and explicit feedback is important.

Attitude, knowledge acquisition and programming skills can be improved by moving to the *changing* level. Probably, it will need more time from the students, because this level is often used in a homework environment. Again, narrative contents and explicit feedback –just in case of using this level together with *responding*– are suggested. Also, some experiences have incorporated advanced features, as high quality contents –this means more work for the teacher– and, good integration with the IDE and advanced programming and visualization interface –this means more development effort if one wants to build her own system–.

Finally, moving to the *constructing* level can improve attitude and knowledge acquisition. The construction process should be effortless, and narrative contents should be added.

Coming from the *viewing* engagement level

This level means low interaction with visualizations. Thus a simple change is to move to the *changing* level, where knowledge acquisition is improved. In addition to narrative contents and explicit feedback, high quality contents have been shown to be effective.

³ both represent a student centered approach rather a high technology centered approach

Moving to the *constructing* level could improve attitude and, as a side effect, knowledge acquisition. Again, the construction process should be effortless, and narrative contents should be integrated in the animations. It can be used together with *presenting* level, improving knowledge acquisition, but students should be free to choose their own construction kits.

5 Conclusions and future work

This is not a meta study, note that we have not included unsuccessful experiences, therefore we can not state if the studied features are significant factors for educational improvements. But we can give some recommendations, we have seen many features present in these successful experiences: narrative and textual contents, feedback to students' answers, and a student centered approach when designing PAV construction kits. Finally, we have identified possible ways to move among engagement levels and its possible effects.

The future work will consider unsuccessful experiences, therefore we will be able to give more formal correlations between engagement levels and educational improvements.

References

- [1] Ahoniemi, T. and E. Lahtinen, *Visualizations in preparing for programming exercise sessions*, Electronic Notes in Theoretical Computer Science **178** (2007), pp. 137–144.
- [2] Ben-Bassat, R., M. Ben-Ari and P. Uronen, *The jeliot 2000 program animation system*, Computers & Education **40** (2003), pp. 1–15.
- [3] Byrne, M., R. Catrambone and J. Stasko, *Evaluating animations as student aids in learning computer algorithms*, Computers & Education **33** (1999), pp. 253–278.
- [4] Crosby, M. and J. Stelovsky, *From multimedia instruction to multimedia evaluation*, Journal of Educational Multimedia and Hypermedia **4** (1995), pp. 147–162.
- [5] Cross, J., T. Hendrix, J. Jain and L. Barowski, *Dynamic object viewers for data structures*, in: *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education* (2007), pp. 4–8.
- [6] Grissom, S., M. McNally and T. Naps, *Algorithm visualization in CS education: comparing levels of student engagement*, in: *SoftVis '03: Proceedings of the 2003 ACM symposium on Software visualization* (2003), pp. 87–94.
- [7] Hansen, S., N. Narayanan and D. Schrimpscher, *Helping learners visualize and comprehend algorithms*, Interactive Multimedia Electronic Journal of Computer-Enhanced Learning **2** (2000), available at <http://imej.wfu.edu/articles/2000/1/02/>, 2008.
- [8] Hübscher-Younger, T. and N. Narayanan, *Dancing hamsters and marble statues: characterizing student visualizations of algorithms*, in: *SoftVis '03: Proceedings of the 2003 ACM symposium on Software visualization* (2003), pp. 95–104.
- [9] Hundhausen, C., *Integrating algorithm visualization technology into an undergraduate algorithms course: ethnographic studies of a social constructivist approach*, Computers & Education **39** (2002), pp. 237–260.
- [10] Hundhausen, C. and J. Brown, *Designing, visualizing, and discussing algorithms within a cs 1 studio experience: An empirical study*, Computers & Education **50** (2008), pp. 301–326.
- [11] Hundhausen, C., S. Douglas and J. Stasko, *A meta-study of algorithm visualization effectiveness*, Journal of Visual Languages and Computing **13** (2002), pp. 259–290.

- [12] Jain, J., J. Cross, T. Hendrix and L. Barowski, *Experimental evaluation of animated-verifying object viewers for java*, in: *SoftVis '06: Proceedings of the 2006 ACM symposium on Software visualization* (2006), pp. 27–36.
- [13] Kann, C., R. Lindeman and R. Heller, *Integrating algorithm animation into a learning environment*, *Computers & Education* **28** (1997), pp. 223–228.
- [14] Kehoe, C., J. Stasko and A. Taylor, *Rethinking the evaluation of algorithm animations as learning aids: An observational study*, *International Journal of Human-Computer Studies* **54** (2001), pp. 265–284.
- [15] Kumar, A., *Results from the evaluation of the effectiveness of an online tutor on expression evaluation*, in: *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education* (2005), pp. 216–220.
- [16] Laakso, M.-J., T. Salakoski, L. Grandell, X. Qiu, A. Korhonen and L. Malmi, *Multi-perspective study of novice learners adopting the visual algorithm simulation exercise system TRAKLA2*, *Informatics in Education* **4** (2005), pp. 49–68.
- [17] Lawrence, A., “Empirical studies of the value of algorithm animation in algorithm understanding,” Ph.D. thesis, Dep. of Computer Science, Georgia Institute of Technology (1993).
- [18] Lawrence, A., A. Badre and J. Stasko, *Empirically evaluating the use of animations to teach algorithms*, in: *Proceedings of the 1994 IEEE Symposium on Visual Languages* (1994), pp. 48–54.
- [19] Moskal, B., D. Lurie and S. Cooper, *Evaluating the effectiveness of a new instructional approach*, in: *SIGCSE '04: Proceedings of the 35th SIGCSE technical symposium on Computer science education* (2004), pp. 75–79.
- [20] Naps, T., G. Roessling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger and J. Velázquez-Iturbide, *Exploring the role of visualization and engagement in computer science education*, *SIGCSE Bulletin* **35** (2003), pp. 131–152.
- [21] Stasko, J., *Using student-built algorithm animations as learning aids*, in: *SIGCSE '97: Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education* (1997), pp. 25–29.
- [22] Urquiza-Fuentes, J., “Semiautomatic generation of educational functional program animations (in spanish),” Ph.D. thesis, Department of Computer Science Languages and Systems - I, Universidad Rey Juan Carlos (2007).
- [23] Urquiza-Fuentes, J. and J. Velázquez-Iturbide, *An evaluation of the effortless approach to build algorithm animations with winhipe*, *Electronic Notes in Theoretical Computer Science* **178** (2007), pp. 3–13.